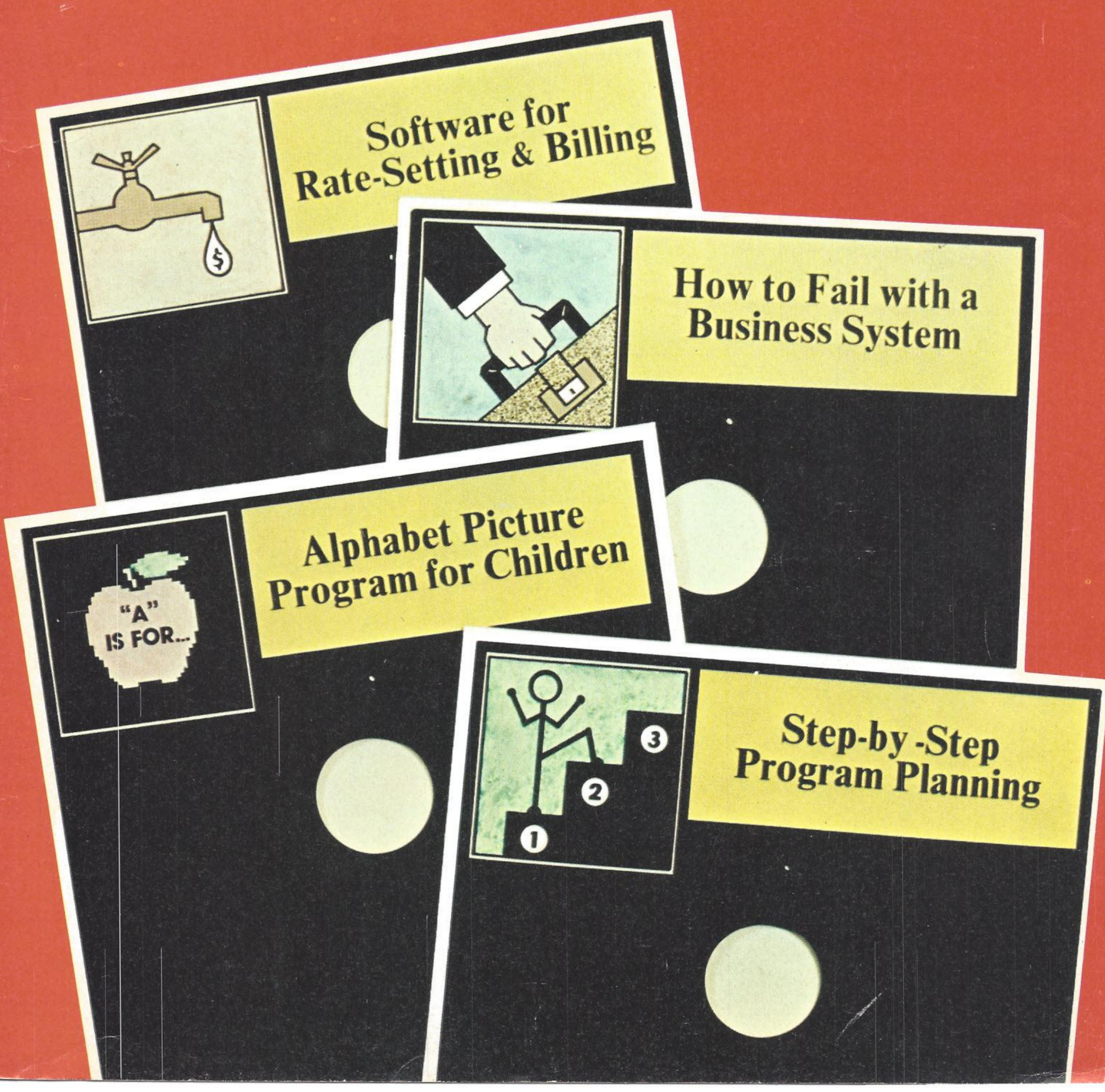
TRS-80 T-BUG • Game-Playing Algorithm • Robots Game

$2.00

Personal Computing

JUNE 1979



CAST A SPELL.
WIN A SORCERER.

If you've written software in Altair Basic, you've written "spells" for the Exidy Sorcerer.

Now, make it pay off!

There's never enough software. Particularly good software.

That's why Exidy is sponsoring a software contest where nobody loses.

Altair programs run on Sorcerer. The Sorcerer computer's Standard Basic is compatible with Altair 4K and 8K Basic. So our contest is open to programs —we like to think of them as "spells" or "Sorcery"—written in all three of those Basic versions.

WIN THIS EXIDY SORCERER.

Trade one of yours for one of ours. Just for entering a program in our contest, we'll send you a new, professionally written and documented program. Free. It's a classic game of concentration that's a fun mind-stretcher for both kids and adults. Plus you'll get our new 20" by 24" color poster.

And maybe 99 more good programs. We'll publish a bound book of the best programs entered—up to 100 of them, with full credit to each author. If you enter you can have a copy for just the printing and mailing cost. And if your program is included, you get the book free.

And maybe a free Exidy Sorcerer: Submit one of the four programs judged "best," and win a free Sorcerer computer. (Or choose Sorcerer accessories of equal value.) There'll be one winner in each of the following categories: Business, Education, Fun & Games, and Home/Personal management.

Test-run your entry free. Take your program to any participating Sorcerer dealer if you want to give it a test run. At the same time, maybe you'll want to jazz up your program to take advantage of Sorcerer's state-of-the-art features. These include 512 by 240 high-resolution graphics; user-defined characters; and dual cassette I/O, among others.

You can turn in your entry right at the dealer's. And collect your poster and new program on the spot.

Enter now. Send us your entry with the coupon. Or visit your dealer. But cast your best spell at Exidy now. And see if you can't make a free computer appear on your doorstep.

RULES:

1) Entries, including documentation, must be printed by computer or typed double spaced on 8½ by 11 paper, with your name on every page.
2) Enter as many times as you like. This coupon, or a copy of it, must be completed and attached to all entries.
3) Enter at any participating Exidy Sorcerer dealer, or mail entries postpaid to the address on this coupon.
4) Entries must be received by midnight, Aug. 31, 1979. Winners will be notified by Nov. 30, 1979. For a list of winners, send a self-addressed, stamped envelope marked "Winners List" to the coupon address.
5) You warrant, by your signature on this coupon, that all program and documentation material included in your entry is entirely your own original creation, and that no rights to it have been given or sold to any other party, and you agree to allow Exidy to use, publish, distribute, modify, and edit it as it sees fit.
6) All entries become the property of Exidy, Inc. No entries will be returned, nor any questions answered regarding individual entries. No royalties, payments or consideration beyond the items set forth in this advertisement will be given to any entrant.
7) Judging will be by a panel of experts chosen by, and including representatives of, Exidy, Inc. Judges may assign programs to whichever entry category they consider appropriate. Decision of the Judges is final.
8) Employees of Exidy, Inc., its dealers, distributors, advertising agencies and media not eligible. Void where prohibited, taxed or restricted by law.

CIRCLE 1

EXIDY, INC.
969 W. Maude Ave.
Sunnyvale, CA 94086

Gentlemen:
Here's my "spell." Send me my free program and poster. If I win, send my Exidy Sorcerer computer to:

NAME ______

ADDRESS ______

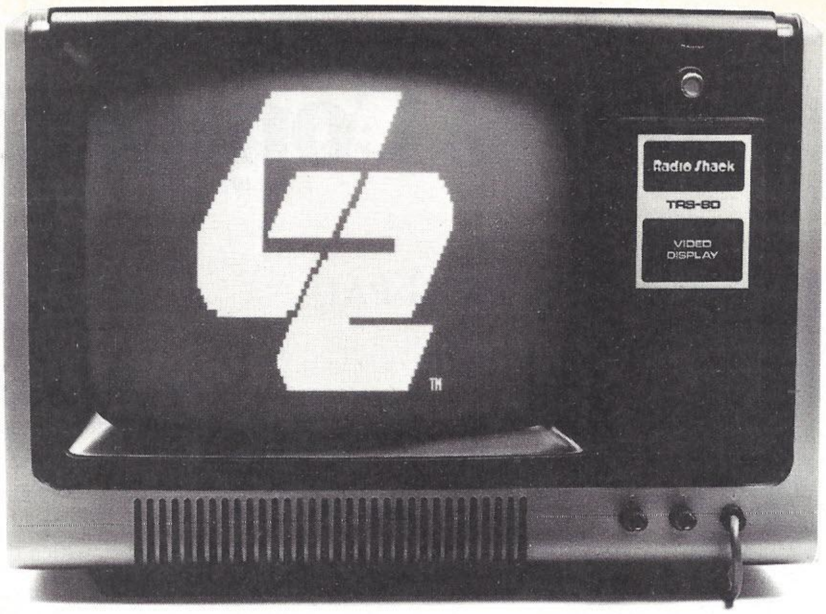
CITY ______ STATE ______ ZIP ______

DAYTIME PHONE ______

TITLE OF PROGRAM ______

CATEGORY ☐ Business ☐ Fun & Games ☐ Education ☐ Home/Personal Management

SIGNATURE ______ DATE ______



INTRODUCING G2 LEVEL III BASIC.

Now do more than ever before with the most powerful Basic you can buy for the TRS-80.

Open the manual and load the cassette. Then get ready to work with the most powerful Basic interpreter you've ever had your hands on...Level III Basic for Radio Shack Computers. It loads right on top of the Level II ROM, and in just 5K of space, opens up your capability to new dimensions. For starters, this new cassette-based interpreter gives you the whole catalog of disk programming power. Plus graphics commands. Plus powerful editing commands. Plus long error messages, hex and octal constants and conversions, user defined functions and a number of commands never before available on either cassette or disk interpreters!

Easier Loading, Fewer Keyboard Errors. G2 Level III Basic eliminates aggravations you've had, including keyboard "bounce" and those super-sensitive tape deck settings. Programs will load easier, and you'll have far less trouble with input errors.

Basic Access to RS-232. Until now, if you wanted to access your RS-232 interface, you had to work in assembly language. G2 Level III Basic does the work for you, letting you use your interface with Basic statements.

Have You Wished for More Power? This new interpreter gives you 10 machine language user calls for subroutines, long error messages, a new TIME$ call for your real time accessory, plus measure or limit input timing that lets you put a time limit on responses when you're playing games or giving exams. And the list doesn't stop here.

Easier and More Powerful Graphics. This new Basic includes three simple commands that can eliminate dozens of program steps. PUT transfers information from a designated array to your screen; GET reverses the process. LINE makes your computer do the work when you input beginning and end points. Give it two diagonally opposite corner locations, and it'll outline the rectangle you're looking for.

Only Microsoft Could Do It. G2 Level III Basic was created by Microsoft, the same company that wrote Level II Basic for Radio Shack. And it actually uses Level II as a foundation for this enhanced add-on. By the time you've mastered all it can do, calling up the flexibility of the graphics commands, and even enjoying the convenience of renumbering, you'll wonder how it was all possible. It's like getting a whole new computer for your computer.

Available Now for Only $49.95. You get the power that might otherwise cost you hundreds of dollars in additional equipment for only $49.95. Price includes the User Manual, a Quick-Reference Card, and a preprogrammed cassette tape. Load the tape, open the manual, and get ready to work with the most powerful Basic Interpreter you've ever had your hands on. G2 Level III Basic for the TRS-80. Another member of the growing G2 Personal Computer Program Library.

For the name of the G2 dealer nearest you, call us toll-free at 800/538-8540 or 800/538-8541. In California, please call 800/672-8691.

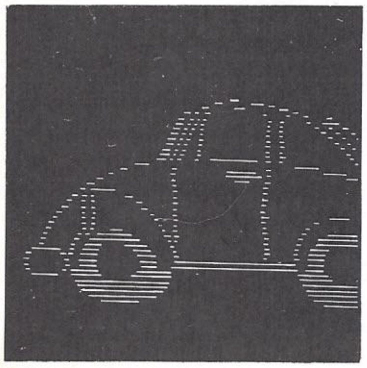
THE REASON YOU BOUGHT YOUR COMPUTER. G2™

GRT® CORPORATION
A Product of GRT Corporation
Consumer Computer Group
1286 North Lawrence Station Road, Sunnyvale, California 94086, 408/734-2910

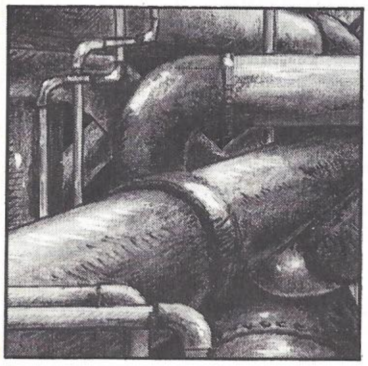
CIRCLE 3

Personal Computing

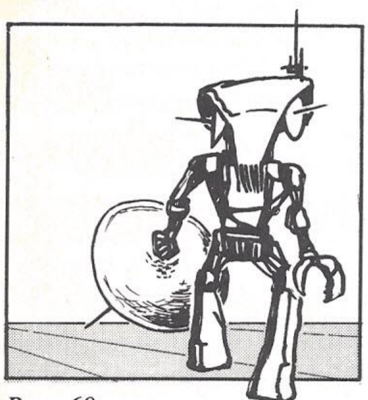
Publication Number USPS 370-770 JUNE 1979 Vol. III No. 6



Page 38



Page 43



Page 60

Cover Design
by Stephen C. Fischer

LAUNCHING PAD

DIGGING IN

IN THE MONEY

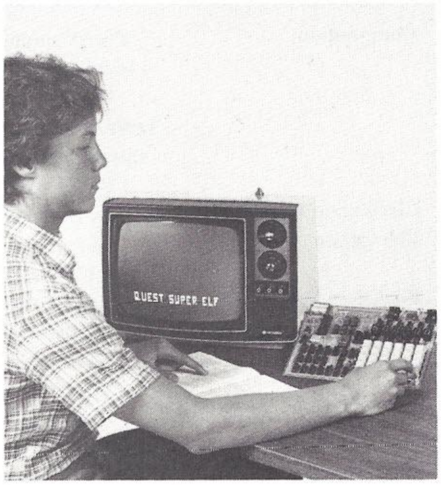
ON THE LIGHTER SIDE

FUTURE COMPUTING

DEPARTMENTS

Business or Pleasure?
(or both!)

Quest provides a very affordable line of computer products beginning with the Super Elf at $106.95 and includes several different microprocessor based systems at various levels up to a sophisticated business system for the small businessman or serious hobbyist.



$106.95 TV based Super Elf.

In the Beginning . . .

The Quest Super Elf is ideal for the beginning computer user and yet is completely expandable to a large computer system. It is easy to use and does so much for so little (under $150.00 complete). Easily connects to your own TV and lets you program animated screen pictures as well as music and games. Good software is available with new programs every month. Give it to a member of your family or a relative. It is not a toy or just a game machine. It will give hours of pleasure while you begin to learn what computers and programming are all about.

The Business System

The Quest Business System is designed around the powerful Z80 microprocessor and the S-100 bus and is more affordable than you might think. It is a very versatile system allowing the user flexibility in cost and performance using the *modular* approach. It can be purchased a step at a time—only what you need, when you need it, beginning with the self-contained desk/cabinet and the computer, adding the floppy disk system and a high speed printer with a smart terminal and CP/M™ operating system. Or, if you already have a printer and a terminal, for example, you may purchase only the other component modules. Of course, the complete system is available tailored to your requirements. Available software packages are standard, including Integrated Business System, Word processing, Basic, Fortran, etc.

Can we put a system together for you?

Need more info?

Write or phone for further details on all computer systems, large and small, for learning or just plain pleasure!

2322 Walsh Avenue, Santa Clara, CA 95051
Tel: (408) 988-1640

Quest ELECTRONICS

CIRCLE 4

Personal Computing

JUNE 1979 Vol. III, No. 6

Publisher/Editor	Harold G. Buchbinder
Managing Editor	Don Wood
Associate Editor	Harry Shershow
Assistant Editor	Marjorie Morse
Editorial Assistant	Joe Bobbey
Art Staff	David Bastille Stephen Fischer Sharon Fisher Josh Randall Sookyun Rha Donni Richmann Susan Ross Richard Sarno
Production Coordinator	Sarah Jewler
Advertising Production	Jonna Yager
Composition	Cheryl Dunphy Julianna Crosier Susan Hull Linda E. Ketchum Claudia Majetich Kathie Willis
Circulation Manager	Dave Harriman
Circulation	Sue Duchin Regina Harrington
Bookstore Sales	Marlene Stibal
General Administration	Esther Shershow Marion Pearlman Sarah Binder Don Schaaf Martha Hawkins
Sales Manager	Arthur A. Daks

Advertising Sales. Northeast: George Palken, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic:** Arthur Daks, Arlene Joseph, Benwill Publishing Corp. 92 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550. • **Mid-West:** Hank Bean, 2663 Hillside Lane, Evanston, IL 60201; (312) 475-7173 • **Northwest:** Ralph Peterson, 1380 Oak Creek Drive, #212, Palo Alto, CA 94304; (415) 328-4392 • **California:** Yuri Spiro, Benwill Publishing Corp., 924 Westwood Blvd., Los Angeles, CA 90024; (213) 478-3017 • **Japan:** Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-ku, Tokyo 166; Phone (03) 311-1746.

Publisher. Published monthly by Benwill Publishing Corp.: Harold G. Buchbinder, Chief Executive Officer; George Palken, President; Domenic A. Mucchetti, Treas. Executive, Editorial and Subscription Offices: 1050 Commonwealth Ave., Boston, MA 02215. Controlled Circulation postage paid at Long Prairie MN. Membership in Audit Bureau of Circulation pending.

Subscription rates.U.S.: 1 year (12 issues) $14; 2 years (24 issues) $26; 3 years (36 issues) $38. Canada & Mexico: add $4/year for surface mail, $8/year for airmail. Other countries: add $8/year for suface mail, $36/year for airmail. Send subscription orders and address changes to: Circulation, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215. In Japan/Asia: Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-ku, Tokyo 166, Japan; Phone (03) 311-1746. In Europe: Personal Computing c/o LP Enterprises, 313 Kinston Road Ilford, Essex, Eng. IG1 1PJ tel: 01 553-1001. **Back Issues.**U.S.: $3. Canada & Mexico: $4. All other countries: $6.

Line Renumbering Renumbered

Dear Sirs:

I just received my copy of the March PC today, and have been proofreading my article, "Line Renumbering on the PET". Unfortunately, I found two errors, one each in the BASIC listing (page 27) and the assembly language listing (page 28). Both are my fault... I'm embarrassed.

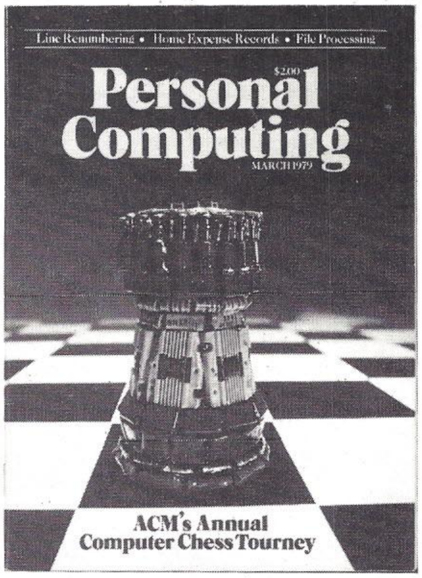
First, a logical flaw: the BASIC line renumberer as given does not, when I test it, correctly renumber lines such as "IF A THEN GOTO 3", because the program skips over the word following "THEN" when it finds that that word is not a number. The solution is simple: instead of line 63580 which now reads "IF N$="" GOTO 63530", replace the line with "IF N$="" THEN L=L-1:GOTO 63530", which prevents the automatic incrementing of L in line 63530 from skipping over an important character or word.

The second flaw is in the listing, not the program, on page 28. Note that line 7502 apparently follows line 7500, a three-byte instruction. This can't be; in fact, the "30" which is in memory location 7502 has been repeated twice. I believe that in my photo listing which I sent you, I must have begun a new frame with 7502 instead of the correct 7503, and through an unfortunate coincidence, the shifted opcodes made an *apparently* legal listing. Sharp readers could probably spot and correct the discrepancy (although they would have to be pretty alert!), but it would be best to give the correct version of lines 7503-7515:

```
7503: 141 255 30        STA 7935
7506: 173 238 30        LDA 7918
7509: 141 254 30        STA 7934
7512:  32 251 29        JSR 7675
7515:     160  0        LDY   #0
```

That's it. As far as I can tell, everything else is correct. Again, I apologize for these slip-ups.

Mark Zimmerman
Pasadena, CA



Barrett's Keybounce Solution

Dear Sir:

You may recall my letter published in the December 1978 issue of *Personal Computing*. In it I reported a bug in Level I TRS-80 BASIC, a bug which may blow up the interpreter when an ON GOSUB or an ON GOTO is nested within a FOR loop. I'm pleased to say that this problem goes away with Level II.

What caused the accidental discovery of "Barrett's Bug" is a TRS-80 glitch which had me tearing my hair out for some time: keyboard chatter. Fellow TRS-80 owners will remember Radio Shack's mention of keybounce in their latest newsletter. It is stated there that multiple letters from one keystroke usually 1) are due to dirty contacts, 2) occur in Level II systems and 3) only appear after some use. Well, sir, I had the problem from the very day I bought my (originally) Level I computer, and I am not alone. Every TRS-80 owner I have questioned has encountered key chatter to some degree or another.

I wrote a letter to Hugh Matthias of Radio Shack Customer Services and asked him how I could remedy the problem. Despite my stamped, self-addressed envelope, he never sent me a reply. So then I tried phoning the number provided in the addendum to the Level II BASIC Reference Manual. Still no luck. I wore my finger out dialing, on different weekdays and for hours at a time. I finally gave up on Fort Worth.

Frustrated but determined, I found a dealer who didn't answer me with a shrug and then head for the CBs. He would have the repair center clean my contacts, for a minimum labor fee of $17.50! (Thinking that keybounce was a Level I problem, I had foolishly let my warranty run out.) However, he could not guarantee that all would be well. My best bet, said he, was to wait for Radio Shack's upcoming debounce program.

I had to do neither, thank Heaven. A friend of mine finally gave me the answer, from a TRS-80 user's group newsletter. (Bless them!) As I had suspected, the problem results from improper construction of the keyboard.

Multiple letters appear when a key's contacts are set too far apart. Take a hook made with a paperclip and pull off the cap of the offending key. Inside you'll see two contacts: a solid one and one with four "fingers". Take a pair of tweezers and clamp the side of the exposed key. At the same time push the solid contact toward the fingered one with a small screwdriver. Press the key cap back on, and your keybounce should be gone. And you didn't even have to open your precious machine and thereby break your warranty!

In just 10 minutes, a problem that plagued me for six months was solved completely. Let me add that I am not a hardware buff. If I can do it, anybody can. The only trick is holding onto the side of the key while you push the contacts closer together. (By preventing the key from depressing, you are adjusting the contacts in a fully open position.)

For the money, the TRS-80 is a fine machine with impressive software. However, more than once did keybounce cause me to consider trading my machine for an Apple II or a Sol. Unless Radio Shack dealers become more willing and competent to deal with "nuisance" problems such as this one, they will lose all but the patient hobbyist.

As for myself, I plan to get my peripherals at local Byte shop. Once burned, twice shy.

Gary L. Barrett
Oaks, PA

Editor's note: We contacted Hugh Matthias, manager of Radio Shack's Computer Support Operations, concerning the problems mentioned in Mr. Barrett's letter. In his reply, Mr. Matthias noted that "any time you are dealing with any mechanical type of contact in a keyboard, there will always be a problem with keybounce." While there have been problems with TRS-80 keybounce, he said, "now that Radio Shack is higher on the learning curve, keybounce is no longer considered to be a 'problem'." Keybounce problems are now fixed at the factory before the computer is shipped.

Mr. Matthias also noted that a software cassette, available free of charge from local Radio Shack stores, eliminates keybounce problems. The tape is named "KBFIX".

"I would like to apologize very deeply to Mr. Barrett for not answering his letter," Mr. Matthias added. "Please feel reassured that I answer as many letters as I possibly can as I can get to them. I really don't understand why he would have any problem getting through on our lines, because we have multiple lines to Radio Shack Customer Services." To reach the Customer Services department, call toll-free (800) 433-1679 or dial direct (817) 390-3583. —*D.W.*

A Word from the Parish on the Pope

Dear Editor:

In a letter to the editor entitled Pope Paul John Paul I, in your February 1979 issue, the author points out very clearly how impractical it would be for successive Popes to follow the practice of combining the names of their two predecessors. The predecessor of Pope Paul VI was John XXIII, not John XXII.

To make this point does not require the use of a computer. However, considering the large size parishes are now, we could make very good use of a microcomputer in a parish. If there are any parishes using a microcomputer, and who would be willing to share their existing programs, I would be very interested to hear from them. Any helpful information regarding computer use in Churches would be gratefuly received.

Reverend Nicholas J. Boomars
Pastor
33333 Mayfair Ave.
Abbotsford
British Columbia V25 IP4
Canada

Ugly BASIC Bugs

Dear Editors:

While reading your March issue I ran across an article in your Random Access section called "Ugly BASIC," Seeing a challenge, I tried it, but couldn't figure it out. So, I went to my TRS-80 and while loading it realized there was something missing from lines 30,60 and 160. Also, you were asking for "Human BASIC" programs. So I took some time out and came up with a program. See if you can find the hidden word:

```
 10  GOTO 180
 20  FOR N=1 to 3
 30  FOR B=1 to 2
 40  IF N > 2 THEN 100
 50  IF B=1 THEN 120
 60  GOTO 80
 70  PRINT "N";
 80  IF N < 2 NEXT B
 90  NEXT N
100  PRINT "E";
110  GOTO 140
120  PRINT "T";
130  GOTO 90
140  Z=0
150  GOTO 200
160  PRINT "T";
170  GOTO 240
180  PRINT "A";
190  GOTO 20
200  PRINT "N";
210  Z=Z+1
220  IF Z > 1 STOP
230  GOTO 160
240  FOR Q=1 to 2
250  IF Q=1 THEN 280
260  PRINT "O";
270  GOTO 200
280  PRINT "I";
290  NEXT Q
300  END
```

Jonathon Cook
Portland, OR

Editor's note: As Mr. Cook notes, information was omitted from three lines in the Ugly BASIC program (Random Access, March 1979). The corrected program reads:

```
 10  GOTO 210
 20  FOR A=1 TO 3
 30  IF A> 1 THEN 50
 40  GOTO 140
 50  FOR B=1 TO 2
 60  IF A < 3 THEN 90
 70  PRINT "E";
 80  GOTO 130
 90  IF A=2 THEN 120
100  PRINT "M";
110  GOTO 130
120  PRINT "T";
130  NEXT B
135  GOTO 160
140  PRINT "O";
150  GOTO 50
160  IF A < 2 THEN 180
170  GOTO 190
180  PRINT "I";
190  NEXT A
200  GOTO 230
210  PRINT "C";
220  GOTO 20
230  END
```

See page 101 for answers to both these programs. —*M.M.*

The Debate Goes On

Dear Editor:

You must have published Mr. Kelly's absurd letter (Feedback, March PC) as a test for your readers, to see who would be the first to discover its numerous logical flaws. I hope that several other readers haven't already beaten me to the draw.

In attempting to disprove the Wexelblat hypotheses (Feedback, December PC) that "few programmers who begin their programming life with a language so limited as BASIC ever significantly extend their ability to make full use of the data and program structuring capabilities of higher level languages learned later," Kelly offers a more general proposition that "only a few people who begin life with a simple language are able or willing to go on to higher languages, whatever higher may mean." Kelly argues that since Wexelblat's hypothesis is only a special case of his own more general proposition (which is shown to be false), then "the narrower hypothesis is also false." Since the narrower hypothesis is implied by the wider, and not the other way around, denying the more general proposition neither proves nor disproves any special case. In formal terminology, Kelly has committed the fallacy of denying the antecedent.

As if realizing that his first line of reasoning is not altogether convincing, Kelly then offers another equally preposterous argument. Even though Wexelblat's hypothesis is quantified by the words "few programmers," Kelly hopes to disprove it with a single counterexample. More remarkable yet, he then re-

NEECO

PROUDLY ANNOUNCES THE NEWEST PET MICROCOMPUTERS BY COMMODORE!

PET 2001 — 32K

LARGE TYPEWRITER KEYBOARDS NOW AVAILABLE!

The PET™ is now a truly sophisticated Business System with the announcement of these Peripherals.

The PET™, incorporated with the Floppy Disk and Printer makes an ideal business system for most professional and specialized fields: medicine, law, dental, research, engineering, toolmaking, printing, education, energy conservation, etc. . . The PET™ Business System as a management tool, delivers information to all levels of Business, previously attainable only with equipment many times more expensive. The PET™ Business System is one of the most cost efficient business tools today. It offers a wide range of applications from logging management strategy in major corporations to organizing accounts and inventory control of small businesses. Here are just a few of the cost-saving uses in the corporation, professional office or small business: stock control, purchasing, forecasting, manufacturing costing, customer records, mailing lists, etc. . . The PET™, Floppy Disk and Printer, a compatible business system at a reasonable price — Take a closer look at these Peripherals . . .

PRODUCT	DESCRIPTION	PRICE	AVAILABILITY
PET 2001 — 4K*	4K RAM	$ 595	IMMEDIATE
PET 2001—8K*	8K RAM	$ 795	IMMEDIATE
PET 2001—16K LARGE KEYS	16K RAM	$ 995	IMMEDIATE
PET 2001—32K LARGE KEYS	32K RAM	$1195	IMMEDIATE
PET 2041	SINGLE FLOPPY	$ 595	JULY-AUGUST
PET 2022 PRINTER	TRACTOR/ROLL	$ 995	JULY-AUGUST
PET 2023	ROLL FEED	$ 850	JULY-AUGUST
PET 2040	DUAL FLOPPY	$1095	IMMEDIATE
8 K RETROFIT ROM KIT		$ 50	IMMEDIATE

* THESE UNITS INCLUDE THE C2N CASSETTE AND SMALL KEYBOARD. THE FULL SIZE KEYBOARD UNITS DO NOT INCLUDE A BUILT IN CASSETTE—ORDER C2N

NEECO IS NOW ACCEPTING $25 DEPOSITS ON THE SYSTEM COMPONENTS OF YOUR CHOICE! PURCHASE ORDERS ARE ACCEPTED IN LIEU OF DEPOSITS. NEECO IS A FULL COSTOMER SERVICE ORIENTED COMPANY. PLEASE CALL FOR ADDITIONAL INFORMATION

THE NEW commodore PET PERIPHERALS!

2040

Dual Drive Floppy Disk

The Dual Drive Floppy is the latest in Disk technology with extremely large storage capability and excellent file management. As the Commodore disk is an "Intelligent" peripheral, it uses none of the RAM (user) memory of the PET™. The Floppy Disk operating system used with the PET™ computer enables a program to read or write data in the background while simultaneously transferring data over the IEEE to the PET™. The Floppy Disk is a reliable low cost unit, and is convenient for high speed data transfer. Due tc the latest technological advances incorporated in this disk, a total of 360K bytes are available in the two standard 5¼" disks, without the problems of double tracking or double density. This is achieved by the use of two microprocessors and fifteen memory I.C.s built into the disk unit. Only two connections are necessary — an A/C cord and PET™ interface cord.

IMMEDIATE DELIVERY

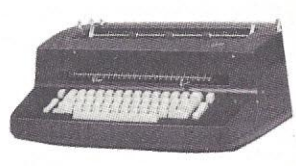
2022

Tractor Feed Printer

The Tractor Feed Printer is a high specification printer that can print onto paper (multiple copies) all the PET™ characters — letters (upper and lower case), numbers and graphics available in the PET™. The tractor feed capability has the advantage of accepting mailing labels, using standard preprinted forms (customized), cheque printing for salaries, payables, etc. Again, the only connections required are an A/C cord and PET™ connecting cord. The PET™ is programmable, allowing the printer to format print for: width, decimal position, leading and trailing zero's, left margin justified, lines per page, etc. It accepts 8½" paper giving up to four copies. Bidirectional printing enables increased speed of printing.

NEW ENGLAND AREA SUB-DEALERSHIP INQUIRIES INVITED!

NEW IBM SELECTRIC II TYPEWRITERS FOR PET! (BRAND NEW IBM II)



PLUG COMPATIBLE VIA USER PORT! WORD PROCESSING SOFTWARE AVAILABLE FROM NEECO. CALL FOR INFO! IBM FOR PET $1995.

$99.95

C2N SECOND CASSETTE

NEECO NOW OFFERS A FULL SIX MONTH WARRANTY ON ALL PETS SOLD BY NEECO! FEEL FREE TO CALL US FOR THE LATEST INFORMATION ON THE NEW PET PERIPHERALS. REMEMBER, A $25 DEPOSIT (OR A PURCHASE ORDER) WILL BACKORDER AND HOLD ANY SYSTEM COMPONENT OF YOUR CHOICE!—WHY NOT BUY FROM THE BEST!—NEECO OFFERS THE BEST IN CUSTOMER SERVICE!—NEECO OFFERS COMPLETE SOFTWARE (COMMERCIAL & CONSUMER) SUPPORT!—MANY PACKAGES AVAILABLE! CALL FOR MORE INFO!—ARE YOU ON OUR MAILING LIST?

NEECO

NEW ENGLAND ELECTRONICS CO., INC.
679 HIGHLAND AVE., NEEDHAM, MASS. 02194
MON.-FRI. 9:30-5:30

(617) 449-1760
MASTER CHARGE/VISA ACCEPTED
TELEX 951021

CIRCLE 31

futes the hypothesis by merely assuming that such a counterexample exists. This is logical reasoning of the most curious type.

There is one other notion implied in Kelly's letter that I would like to comment on. To suppose that even the most powerful artificial programming or mathematical languages are "higher" or more complex than even the most primitive natural languages is a concept which I find very strange indeed. If the complexity of a language is measured by either the number or depth of ideas which can be expressed in that language, then there is really no comparison between programming and natural languages. For example, every syntactically correct FORTRAN statement can be expressed verbally in an unambiguous (though perhaps cumbersome) English sentence. However, I know of no programming language which can handle abstract concepts such as love or justice.

David F. T. Kretzmann
Ft. Meade, MD

Basic Without Style

Dear Editor,

I agree with most of John Nevison's remarks in his article "BASIC with Style" (February 1979). But there are times when you will intentionally want to break his rules. If the BASIC program is to serve as a puzzle, you may want to disguise your program, and possibly even intentionally lead your reader astray. Dr. Ed Dolney and I had this in mind when we wrote the program CHAL1. It certainly does not obey Nevison's rules.

Perhaps your readers will enjoy figuring this program and others out. All of the programs in Figure 2 work properly on a PDP-8 computer with Educomp BASIC. But they should work on most BASICs with appropriate changes. For example, the reader will have to figure out what the "!" and ":" do in the program CHAL1, and make the appropriate changes for his BASIC.

After figuring out how the two FLEE programs work, your reader could generalize those programs to develop his own block letter program. The intent in CHAL2 is to determine how your computer treats logical quantities. Many BASICs have the capability of using logical variables and constants, even though that capability is often not documented.

James L. Boettler
Orangeburg, SC

Editor's note: Printouts from Mr. Boettler's programs are on p. 104.
—*M.M.*

```
FILES COPIED:
CHAL1.BA
   1 REM 'CHAL1' CHALLENGE PROGRAM IN NEWSLETTER
   2 REM J.L.BOETTLER    10/13/77
  10 P   =8:S=   3
  20 !!!KNIFE = QUIT!!!
  30 FORK = STOP
  40 !!!SPOON = HALT!!!
  50 PRINT K!LL
  60 NEXT K!LL NEXT
  70 END!DEAD!END

CHAL2.BA
   1 REM 'CHAL2' CHALLENGE PROGRAM IN NEWSLETTER
   2 REM J.L.BOETTLER    10/12/77
  10 INPUT A,B
  20 D=C=B
  30 E=(A=B)
  40 PRINT A;B;C;D;E, A<B; A=B; A>B
  50 GOTO 10
  90 END

FLEE1.BA
  10 REM 'FLEE1' J.L.BOETTLER
  20 FOR K = 1 TO 7 : READ A : FOR L = 1 TO 6
  30 B = A : A = INT(A/2)
  40 IF B = A*2 THEN PRINT "*          ";
  50 IF B <>A*2 THEN PRINT "*****      ";
  60 NEXT L : PRINT : NEXT K
  70 DATA 61, 0, 0, 45, 0, 0, 62
  80 END

FLEE2.BA
   1 REM 'FLEE2' J.L.BOETTLER 10/13/77
   2 N = 111111
   3 PRINT N,1,N,N,N,N
   4 PRINT 1,1,1,1,1,1
   5 PRINT 1,1,1,1,1,1
   6 PRINT N,1,N,N,1,N
   7 PRINT 1,1,1,1,1,1
   8 PRINT 1,1,1,1,1,1
   9 PRINT 1,N,N,N,N,N
  10 END
```

NCC's Personal Computing Festival

NCC '79

This year's Personal Computing Festival at NCC in New York will concentrate on both the hobbyist and business market, providing a balanced view of low-cost computing, according to Richard Kuzmack, director of the June 4-7 event.

More than 25 technical and applications sessions will cover personal, hobby and business computing. Sessions of special interest include.

Monday (3:30-5:30): The Coming Small Computer Earthquake; session leader: Burchenal Green

Tuesday (8:30-12:30): Small Business Systems: What Can I Get For $ . . . ; session leaders: Dan Ring & Bob Redmond

(10:30-12:30): Small Business Systems: Experiences With Microsystems; session leader: Roger Berger

(8:30-10:30): The Personal Computer in the Schools; session leader: Lou Frenzel

(10:30-12:30): The Personal Computer in Home Education; session leader: Karl Zinn

Wednesday (8:30-4:30): Personal Computing as an Aid to the Handicapped; session leaders: Les Solomon & Larry Hedges

(8:30-12:30): Legal Aspects of Personal Computing; session leader: Harold Novick

Thursday (8:30-12:30): Simulation, Modeling & Games; session leader: Stephen Smith

(10:30-12:30): Personal Computing in Other Hobbies; session leader: Harold Buchbinder

Remaining sessions are:

- Personal Computing Pioneers; Sol Libes
- Inter-Computer Communication; Charles Judice
- Ethics and Crime; Oliver Smoot
- Meeting of Computer Retailers Association; Portia Isaacson
- Personal Computing in Investment Analysis; Reid M. Shay
- Art and Graphics; Bill Etra

- Music; Carl Helmers
- Personal Computing as a Tool in Other Professions; George Cohen
- Computer Speech Synthesis and Recognition; Jim Boddie
- Computer Retailing; Alan P Hald
- Personal Computers: Where Do We Go From Here?; Jack M. Nilles
- Recreational Mathematics; Abraham Waksman
- Personal Computer Maintenance; William J. Schenker
- Hardware and Software Symposium; Ward D. Maurer
- Video Theatre; Taylor Barcroft

Last year approximately 20,000 people attended the Festival, according to Tom C. White of Afips, and a comparable number are expected for 1979.

Although a mixture of business and hobby computerists attended last year's Festival, Kuzmack noted an increase in the number of individuals just getting interested in computers. Many people who went had heard about the growing computer market but were not yet involved with computers themselves.

A new feature of the 1979 Festival is the "hands-on" room containing equipment from manufacturers exhibiting elsewhere in the conference, said Kuzmack. Visitors will have a chance to view and operate up-and-running systems with the help of knowledgeable attendants.

The Personal Computing segment of the NCC has not always been a separate conference, Kuzmack noted. In 1975 NCC in Anaheim, CA, contained few personal computing exhibits. The 1976 conference in New York offered a full day of personal computing sessions but was still within the NCC itself. NCC '77 in Dallas recognized, for the first time, personal computing as a separate part of the conference, but it wasn't until 1978 that the Festival actually became a conference-within-a-conference.

Radio Shack Computer Customer Service

A new, free customer service department has been opened by Radio Shack to solve problems stumping any owner of a TRS-80. Declaring its purpose to be a service of assistance to the novice, the Tandy Company says its telephone lines open from 8:00 a.m. to 5:00 p.m. (Central Standard Time). The department may be called, on a toll-free WATS line, from anywhere in the United States except Texas (location of this new department). The toll-free number is 1-800-433-1679. The regular number for Texans (and others who get busy signals but need quick responses) is 1-817-390-3583.

The computer-customer service group is headed by Hugh Matthias. More than 20 professional people staff this new office, including operators, software consultants and hardware technicians. The company feels that in this way they can support their customers, either future or present, who have questions which cannot be answered by local Radio Shack Customer Service Departments. Tandy Company urges TRS-80 owners to first try the local Radio Shack, which usually can supply the answers to sticky questions. But if additional support or further clarification is needed, then don't hesitate to use the toll free number, says the company. Customers, who are not in a hurry for answers, may prefer to write to the new department at: Radio Shack Computer Customer Services, 205 NW 7th Street, Fort Worth, TX 76106.

Getting MEAN

Decreasing costs of mini and microcomputers are creating opportunities for computer use in school systems, Charles Blaschke, President of Education Turnkey Systems, noted recently. To facilitate software development and dissemination of microcomputer applications in education, the Microcomputer Education Applications Network (MEAN), has been formed. "MEAN is an organized effort to break the software bottleneck of the previous decade," said Blaschke, a veteran of the Computer-Based Education movement which began in the early sixties.

School officials, individuals or groups can join MEAN at no cost. Users of the software systems will receive discounts on purchases of software; developers will receive royalties for useful and practical applications adopted by other schools or districts; and interested MEAN members will be provided with current information on microcomputer technology and applications designed for educational institutions, Blaschke noted.

"Survey research findings already indicate increasing trends, including a greater increase in combined administrative and instructional applications at building levels; increases in games and simulations; greater usage in test scoring, reporting and analysis; and other applications which reduce paperwork and extend the capabilities of teachers and other building staff. Public education has usually been six to eight years behind industry in computer applications; there is now a unique opportunity for a great leap forward in computer technology in education," said Blaschke.

For additional information, contact MEAN, 1030 Fifteenth Street, N.W., Suite 800, Washington, D.C. 20005.

Computer Education for Everyone

School children and executives can both benefit from personal computing. And, these individuals have an opportunity to learn more about what computers can do through programs developed especially for them.

A new television special entitled *"Don't Bother Me, I'm Learning" — Adventures in Computer Education,* will present a view of the microelectronic revolution and its impact on the learning process of children, according to One Pass, Inc., of San Francisco, developers of the show.

The special, viewed through a child's perspective, will deal with computer-based education of elementary school children from the acquisition of basic skills to playing fantasy games.

Participants in the show include Art Luehrmann and educator Dean Brown, who talk about the value of the computer in society; teachers who share what they've seen classroom computers do for their students; parents who've watched their children converse with the computer in math exercises; and children who talk about their "companion".

"BASIC: A Computer Language for Executives", is a two-day course being held July 19 to 20 at the American Management Association's headquarters in New York City.

The course is designed to introduce computing as a management tool, according to course organizers. Topics include Executive Computing: problem solving, planning, forecasting and database systems; Programming Fundamentals: the mindless computer, sequence, decision and iteration, computer languages and BASIC.

Participants will actually work with a computer, performing simple programming.

For more information on the AMA program, contact American Management Associations, 135 West 50th St., New York, NY 10020; (212) 586-8100.

"Chess Challenger-10 Wins Microchess Tourney"
—Personal Computing Magazine
February, 1979

Genius Offspring

"Its successor, the new Chess Challenger "7," is infinitely more powerful."
—S. Samole
President, Fidelity Electronics

Chess Challenger-10 did more than win the Penrod Memorial Microchess Tournament, it literally trounced all opponents. Personal Computing Magazine, February, 1979, reports, *"Chess Challenger-10 emerged as the easy victor with ten wins, two draws and no losses."*

All Top Name Performers

There were no amateurs in the championship playoff. Every contender bore the brand of a well-known electronic chess game, and each was accompanied by its entourage of coaches, programmers, and engineers. After each contestant had played all of the opponents in round robin fashion, the brilliant Challenger-10, stood far ahead of its second place runner-up.

Nobody Knew

Unknown to the other companies, the undefeated tournament leader was being retired after the contest. Taking its place was a far more powerful chess computer, the Challenger "7." This new micro-computer had already beaten the official undefeated champ during a series of pre-tournament warm-up games at the factory. Its engineers explain that it is simply 14 months ahead in technology, in finer algorithm sophistication and in its superb performance.

Improve Your Game to Near Brilliant

Within its seven different levels of play, you can enjoy every degree of chess competition, from beginner to tournament skill. Its total flexibility lets you change games midstream or switch sides with the computer to see how it would handle your dilemna. You can add pieces to your side or take away the computer's Queen. It is a superb teacher!

Touch the PV key and the "7's" total recall memory will verify every piece position on the board. You can even set up hypothetical encounters to test its reaction at each level.

Fidelity's Challenger "7" is able to analyze over 3,024,000 board positions. It masterfully handles over one thousand book openings and will respond to any deviation. Academic openings as Sicilian, French, Ruy Lopez and Queen Gambit Declined, are just some of the challenges to keep you on your toes.

It Knows Every Rule in the Book

The Challenger "7" will permit you to castle or perform an En Passant capture or do so itself, if that is its best move. When your pawn has reached the eighth rank, it will be automatically raised to a Queen, unless you tell the computer to promote it to another piece. It will take on any player and sharpen his skills considerably... but it won't permit illegal moves.

Sound Signals

You just cannot believe the chess I.Q. of this phenomenal unit. The Challenger "7" has a beep audio feedback which sounds to acknowledge your move, and double beeps when the computer has made its response. At Level 1, its average response time is 5 seconds. At Tournament Level 7, the Challenger makes championship decisions in just 3 minutes.

Unbeatable in Price As Well As Play

Best of all, the Chess Challenger "7" is the most affordable electronic unit you can own. It is just $89.95 complete with Staunton designed pieces and UL approved 110V AC adaptor.

All pieces are magnetized, to stay where you place them on the permanent metal board. The set is mounted in a simulated wood-grained housing which measures 12⅛" x 8" x 1." Bright, one-half inch tall LED electronic digits, provide unmistakably clear readout. The unit is backed by a 90-day manufacturer's limited parts and labor warranty.

Enjoy It for 10 Days—At Our Expense

As a gift or for yourself, the "7" is unquestionably the finest chess computer you can select... but, if within 10 days, you are not satisfied, simply return it for a prompt, no-questions-asked refund.

Final Results

Reprinted Courtesy of Personal Computing, February, 1979. P. 66. (Darker lines ours.)

	CONTESTANTS		#1	2	3	4	5	6	7	8	9	Games Won	Games Drawn	Games Lost	FINAL SCORE	FINAL POSITION
			OPPONENTS													
1	MICRO-CHESS 1.0 (Heath H-8)	W		½	Ø	1	Ø	Ø	Ø	—	—	1	3	8	2½	7*
		B		½	½	Ø	Ø	Ø	Ø							
2	MICRO-CHESS 1.5 (TRS-80)	W	½		½	½	Ø	Ø	Ø	—	—	Ø	5	7	2½	6*
		B	½		Ø	Ø	Ø	½	Ø							
3	MICRO-CHESS 2.0 (PET)	W	½	1		1	Ø	Ø	½	—	—	3	4	5	5	4
		B	1	½		½	Ø	Ø	Ø							
4	CHESS CHALLENGER (3 Level)	W	1	1	½		Ø	½	½	—	—	2	5	5	4½	5
		B	Ø	½	Ø		Ø	Ø	½							
5	CHESS CHALLENGER (10 Level)	W	1	1	1	1		1	½	—	—	10	2	Ø	11	1
		B	1	1	1	1		1	½							
6	BORIS	W	1	½	1	1	Ø		1	—	—	7	2	3	8	3
		B	1	1	1	½	Ø		Ø							
7	SARGON I (TRS-80)	W	1	1	1	½	½	1		—	—	6	5	1	8½	2
		B	1	1	½	½	½	Ø								
8	ATARI *Did not play*	W/B	—	—	—	—	—	—	—		—					

* Note: Microchess 1.5 wins 6th place over Microchess 1.0 by virtue of the tie breaking analysis of relative strength of opponents

Animation Easier With Computers

The Swedish Broadcasting Company (SBC) is using a computer to help produce animated films for its television services.

Conventional methods of preparing animated films are enormously time-consuming. Most of the work is very repetitive since each second of finished film requires from 20 to 25 similar drawings. The new system utilizes a Sperry Univac 1100/11 computer with a technique called ANTICS, developed by Alan Kitching, an animation and data processing specialist who manages Grove Park Studios in Camberwell, London, England.

A basic drawing is prepared and entered into the computer using a special light pen. With command words and coded direction, speed and position specifications are also inputted into the computer. The drawing can be modified in different ways; it can be shrunk, enlarged, panned, skewed, shaken or reversed. It can also rotate, jump or rock. The system now contains some 40 commands but Kitching is working on further expansion.

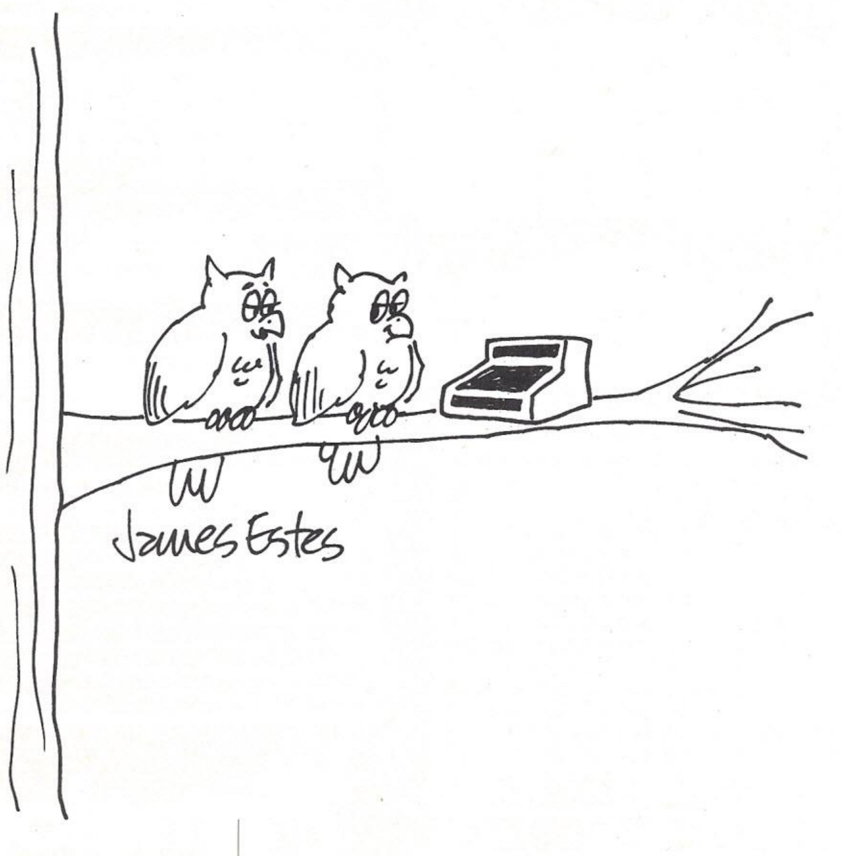
Currently, more than 100 separate color picture elements can be manipulated in a single sequence to move in different ways.

One of the newer concepts within ANTICS is the Skeleton command. In response to a Skeleton command a part of the basic drawing — such as a human character — will move in a natural manner. Using single skeletal figures, a person can specify a pattern of movements based on key positions.

The computer then automatically creates all of the intervening pictures needed to complete the overall sequence of movements and the final result appears as a smooth natural motion which matches the position of the skeletal figures entered into the computer.

When the entire sequence has been created it's presented on a graphic display screen and then recorded on video tape so it can be played back and corrected.

Sven Hjort, Systems Technology Manager for SBC, is enthusiastic about the opportunities for the ANTICS system. "Animated film offers numerous advantages on television in complicated situations that would be impossible to explain or illustrate. With the new system, sophisticated films that would have taken more than six months to produce by conventional means can be made in a few weeks," said Hjort.



"SO THAT'S WHY YOU'RE SO WISE... YOU HAVE A COMPUTER."

Program-of-the-Month Club?

Software best-seller lists and "program-of-the-month" clubs could be in the offing as the demand for prepackaged computer programs soars. An Arthur D. Little, Inc., report citing the rapid growth of the personal computer industry suggests a tremendous opportunity for the "electronic publisher" who can serve the program needs of consumers and small businesses, currently prime users of personal computers. Unlike the early hobbyists who had the desire and the technical skill to produce their own programs, the new breed of user wants a ready-made software package, the report said.

Prepared by Arthur D. Little Impact Services Company, the report indicates that in addition to electronic publishing companies, traditional publishers, software houses and computer manufacturers will be vying for a share of the market.

MAD HATTER SOFTWARE

PROGRAM	DESCRIPTION	TRS-80 L1/2	TRS-80 L2	APPLE	PET	PRICE
GALACTIC BLOCKADE RUNNER	• AN EXCITING SPACE WAR GAME WITH GRAPHICS	•		•	•	$ 9.95
SCI-FI GAME SAMPLER	• 3 GAMES—LUNAR LANDER—STAR MONSTER—SPACE BATTLE	•		•	•	$ 7.95
R/T LUNAR LANDER	• A REAL TIME LUNAR LANDER WITH GRAPHICS		•			$ 7.95
MICRO-TEXT EDITOR	• FORMAT TEXT—SAVE & LOAD TO TAPE—OUTPUT TO PRINTER		•			$ 9.95
OTHELLO III	• A STRATEGY BOARD GAME—PLAY AGAINST COMPUTER OR OTHERS	•		•	•	$ 7.95
AIR RAID	• A REAL TIME, ARCADE TYPE SHOOTING GAME IN MACH. LANG.	•				$ 14.95
MICRO-CHESS	• PLAY CHESS WITH YOUR COMPUTER—VARIOUS LEVELS OF DIFF.	•		•	•	$ 19.95
BRIDGE CHALLENGER	• DON'T WAIT FOR OTHERS TO PLAY—YOUR COMPUTER'S READY	•		•	•	$ 14.95
APPLE 21	• BLACKJACK WITH HIRES GRAPHICS			•		$ 9.95
STAR WARS/SPACE MAZE	• SCI-FI GAMES FOR THE APPLE			•		$ 12.95
RENUMBER	• RENUMBER YOUR BASIC PROGRAMS—RENUMBERS EVERYTHING	•		•		$ 14.95
DISK RENUMBER	• SAME AS ABOVE, BUT ON DISK	•				$ 19.95
PILOT 2.0	• THE EDUCATIONAL LANGUAGE, IN MACH. LANG.—INC. EDITOR		•			$ 14.95
PILOT 3.0	• THE DISK VERSION OF THE ABOVE		•			$ 24.95
APPLE TALKER	• YOUR APPLE SPEAKS! NO NEW HARDWARE REQUIRED			•		$ 15.95
APPLE LIS'NER	• SPEECH RECOGNITION THE EASY WAY—GREAT WITH THE TALKER			•		$ 19.95
TIC-TAC-TALKER	• TIC-TAC-TOE USING SPEECH SYNTHESIS AND RECOGNITION			•		$ 19.95
FORTRAN	• FOR THE TRS-80—SEE MICROSOFT'S ADS		•			$325.00
SYSCOP	• MAKE BACKUP TRS-80 SYSTEM TAPES THE EASY WAY		•			$ 9.95
ANDROID NIM-2	• GAME OF NIM WITH ANIMATED ROBOTS AND SOUND		•			$ 14.95
SNAKE EGG	• A BETTING GAME WITH ANIMATED SNAKES AND SOUND		•			$ 14.95
LIFE 2	• 100 GEN. PER MIN. LIFE & BATTLE OF LIFE W/ANIMATION & SOUND		•			$ 14.95
DCV-1	• PUT SYSTEM TAPES ON DISK EVEN IF IN SAME MEM AS DOS		•			$ 9.95
MUSIC MASTER	• ENTER SHEET MUSIC—THE TRS-80 THEN COMPILES & PLAYS IT		•			$ 14.95
DISK MUSIC MASTER	• SAME AS ABOVE BUT ON DISK W/MANY SELECTIONS		•			$ 24.95
TRS-80 CP/M	• OPENS UP THE WHOLE WORLD OF CP/M SOFTWARE TO THE TRS-80		•			$150.00
PET ASTROLOGY	• DOES ALL THE COMPUTATIONS FOR YOU				•	$ 14.95
PERSONAL FINANCE PKG.	• KEEPS TRACK OF CHECKS, BUDGET, COMPUTES INTEREST	•		•	•	$ 9.95
TRS-80 DISK LIB. "A"	• 9 PROGRAMS ON DISK—INC. BLOC. RUN, OTHELLO ETC.		•			$ 39.95

BUSINESS AND APPLICATION SOFTWARE—CALL OR WRITE FOR DETAILS
16K MEMORY UPGRADE FOR TRS-80, APPLE & SORCERER COMPUTERS — $ 94.95
C-10 HIGH QUALITY DATA CASSETTES W/SCREW HOUSINGS — 10/$ 10.00
VERBATUM 5-1/2 INCH BLANK DISKETTES — 10/$ 34.95

10% OFF IF YOU ORDER 3 SOFTWARE PACKAGES OR MORE **SEND FOR FREE CATALOG—GIVE TYPE OF COMPUTER**

AVAILABLE FROM THESE FINE MICRO COMPUTER DEALERS

J&J ELECTRONICS LTD
28 COLLINGTON AVE.
BEXHILL-ON-SEA, E. SUSSEX, ENG.

HOBBY WORLD ELECTRONICS
19355 BUSINESS CENTER DR. #6
NORTHRIDGE CA 91324

CAPITOL COMPUTER SYSTEMS
3396 EL CAMINO AVE.
SACRAMENTO CA 95821

COMPUTER CABLEVISION, INC.
2617 42ND ST. NW #2
WASHINGTON DC 20007

COMPUTER VILLAGE
931 SW 87TH AVE.
MIAMI FL 33174

AUGUST AUTOMATION
28 MILK ST.
WESTBORO MA 01581

TRS-80 SOFTWARE EXCHANGE
17 BRIARCLIFF DR.
MILFORD NH 03055

KENNEDY SYSTEMS
74 BROAD ST.
LYNDONVILLE VT 05851

L. C. SALES
100 HINCHEY AVE. #705
OTTAWA, ONT., CAN. K1Y4L9

THE CPU SHOP
39 PLEASANT ST.
CHARLESTOWN MA 02129

OP AMP TECH BOOKS
1033 N. SYCAMORE AVE.
LOS ANGLES CA 90038

ADVANCED COMPUTER PRODUCTS
1310B E. EDINGER
SANTA ANA CA 92705

TO ORDER BY PHONE OR FOR DEALER INFO—CALL—(617) 682-8131
ADD 75c SHIPPING & HANDLING • MASS. RESIDENTS ADD 5% SALES TAX
MAD HATTER SOFTWARE • 900P SALEM RD • DRACUT, MA 01826

VISA

CIRCLE 9

Bubble Memory Bursts

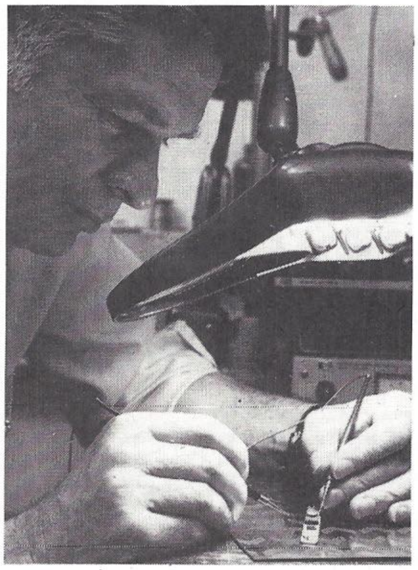
Bell Laboratories is predicting a new era for magnetic bubble technology, with much faster, cheaper and more compact bubble memory, said Andrew Bobeck, a Bell Lab scientist.

Magnetic bubble technology is now being used in the Bell System to store digitally recorded messages such as "We're sorry, but the number you have reached has been changed to. . ."

Improved bubble memories may be used in the telephone system for high-capacity switching systems which direct one telephone call to another, and in a variety of terminal equipment, especially those using microprocessors.

Magnetic bubbles are tiny cylindrical magnetized areas — less than 1/16 the diameter of a human hair — contained in a thin film of magnetic materials such as a garnet layer. These areas can be moved about electronically so they can be used for computing and storing information. Since their invention, magnetic bubbles have showed promise as a reliable, low-cost alternative to other mass memory devices: magnetic tapes, floppy disks and semiconductor memories.

The new era, according to Bobeck and four colleagues, will involve devices markedly different from the magnetic bubble techniques in use today. Today's devices enclose the garnet material within a pair of coils. These



coils are used to generate a magnetic field which moves the magnetic bubbles in the garnet film.

Inherent in today's bubble technology were a number of speed, size and memory-storage limitations which Bell Labs engineers have succeeded in overcoming. The new bubble developments replace the present coil drive with one or more wafer-thin conducting layers. These have a pattern of oval-shaped holes etched in them, and an applied current that moves the magnetized bubbles. Because the coils are eliminated, the magnetic bubble device is reduced one-third in size and requires much simpler drive electronics. In addition, the bubbles are stepped ahead more than 10 times faster than before.

Perhaps one of the most important features of the manufacture of bubble memories is the compatibility of the new developments with semiconductor processing technology. Many of the same manufacturing techniques developed for semiconductors can be applied directly to magnetic bubbles. The years of semiconductor experience can be readily transferred to the emerging bubble field.

The compatibility of these two important technologies should have a profound impact on the telephone and microelectronics industries, both of which are constantly seeking faster, cheaper and denser memory technologies, concluded Bell Labs.

Quick Access to Realty Listings

Computers in Phoenix, AZ, are giving realtors an edge in obtaining updated real estate listings. A system developed for Multiple Listing Service members is providing 55 firms in the area with instantaneous access to new listings.

"In the past, real estate salespeople have had to wait for up to six days to receive a book telling them what new properties had come up for sale," said Paul Behrendt, MLS spokesman. "With a computer terminal in their office connected by telephone line to the main computer, the listing is available immediately, giving realtors about a week's lead time over their non-computerized competitors."

The KMI Insta-Data Realty System is a joint venture between J. Keith Gunville, president of the firm, and Messenger Corporation, a graphic arts organization.

"The terminals are being installed as fast as the telephone lines become available," Gunville said. "I expect the system to reach 250 units by December of this year."

The heart of the system consists of twin 128-K Varian Data V-76-77 computers located at the KMI complex. As new listings are received by MLS and forwarded to KMI, they are fed into the computer. In the past this data base was used only to set the type for the weekly MLS listing book. Now, members with terminals can tap directly into the system, add or change their own listings, and have access to any listing days before the book is published.

"A realtor with a terminal may be able to sell a home listed on the computer before the book is even published," Gunville said.

The computer does more than just show listings, Gunville noted. It locates homes for sale within a particular area such as a school district, or selects homes with options like swimming pools, fireplaces, dining rooms or horse privileges.

Baseball Scouts and Computers: A Team Effort

Major league baseball scouting isn't what it used to be. Today, information on a prospect's hitting, throwing, running and fielding abilities is just as likely to come from a computer as from the pen of a retired player working as a scout for his old club. In fact, 17 of the 26 major league teams subscribe to the computerized scouting services offered by the Major League Scouting Bureau in Newport Beach, CA, according to officials.

Headed by former player-general manager Jim Wilson, the MLSB employs 60 professional scouts who watch baseball games across the country on behalf of their clients in both the National and American Leagues. The information they gather is stored in Honeywell's Computer Network Service, Datanetwork. "Scouting personnel," said Wilson, "are baseball people who don't want to know any more about computers than necessary. Honeywell designed its system so that non-technical people like us would have no trouble using it."

Datanetwork enables MLSB clients to request individual player scouting reports as they become available "at less cost than the teams could do it for themselves," said Wilson.

The large data base containing the player performance information is updated daily from the Major League Scouting Bureau's officers in Newport Beach. The client teams, using either video-display or teletype terminals, simply dial into Honeywell's large-scale computer system in Minneapolis and request the profiles of players by name, position, location or ranking.

"Scouting is a year-round activity," Wilson says. "Even in the off-season, we're using Datanetwork to do our homework before spring training camps open, storing data from observations of some players in clinics and getting our files ready for the January draft."

During the season, scouts in five regions cover games played by high school, college and minor league teams. They rate promising players by height, weight, ability to play their position and batting skill.

These ratings are combined with subjective comments on the players, then sent to the MLSB main office where they're loaded offline onto a cassette, and dumped online daily into the computers. Dial-in access and simplified procedures allow non-technical users to process all this information.

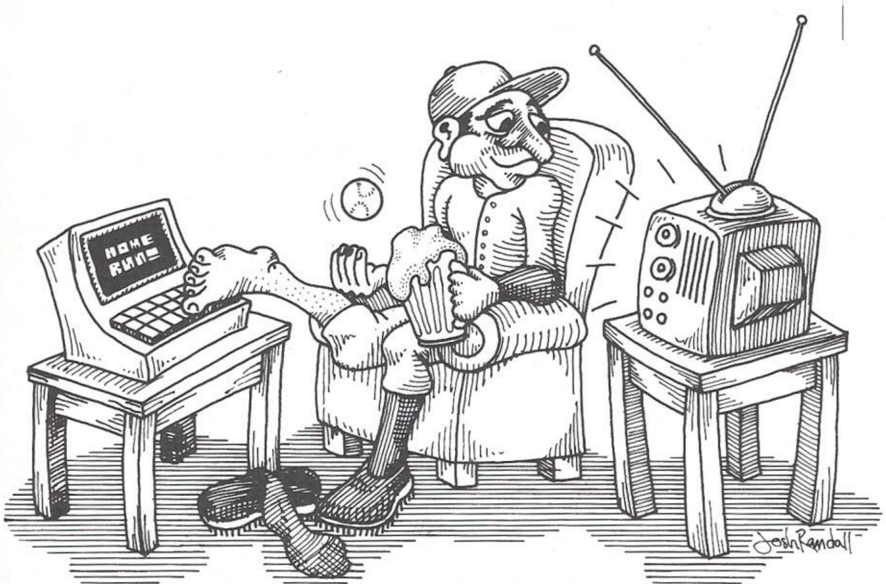
"Twice a year, we produce reports for each team on the 500 most promising high school and college kids and mail them to the scouting director before the free-agent draft in January and June," Wilson said. A similar report — much larger and more comprehensive — on all players in the minor league pro system is produced just after the draft in June and sent to each subscribing team, then updated throughout the minor league season.

A printout on a particular player will include the latest preferential order relative to other players contained within the data base. The preferential order is based on all information available from scouts, who use a 2-to-8 rating scale for each category of performance.

Preferential lists for the free-agent drafts contain all the players by position, scout and state where they played. To make this list, players must have a total of 20 points minimum, with 80 points the maximum.

Reports for the pro system are larger and more complicated. They assess 23 different aspects of a player's performance, including running, throwing, hitting, fielding, accuracy, range and aggressiveness. An overall evaluation from all scouting reports is produced by the computers and added to the composite comments.

The reports are printed and arranged according to the farm system in which the players play. Only those players in a particular farm system are profiled for the parent major league club. Password and other identifiers are used to limit access to specific teams for information on individual pro system players.



Club News

Several additions and changes have been reported from computer clubs and users groups.

Computer Mart of Massachusetts has formed the Sorcerer User's Group with Bruce McGlothlin, coordinator, and John Vachon, software engineer. A monthly newsletter, "The Exidy Monitor", is offered to members. Membership fee is $5 annually. For more information contact Bruce McGlothlin at Computer Mart of Massachusetts, Inc., 1395 Main St., Waltham, MA 02154; (617) 899-4540.

A new TRS-80 users group has been started in Alabama. The Central Alabama TRS-80 Computer Society plans to provide a newsletter and form a library. Meetings will be held the third Tuesday of each month. For more information contact Walter F. Bray, 2073 Rexford Rd., Montgomery, AL 36116.

The new Abacus (Apple Bay Area Computer Users Society) meets the second Monday of each month at the Hayward Byte Shop, 1122 B Street, Hayward, CA. For more information call Ed Avelar, president, at (415) 583-2431; or David Wilkerson, secretary, (415) 482-4175.

The American Association of Physics Teachers has announced its first national meeting with the theme: the use of microcomputers in physics teaching. The meeting will be held at New Mexico State University in Las Cruces, NM, June 19 to 23. Activities include a one-day workshop on an "Introduction to Microcomputers"; a symposium devoted to the role of microcomputers in physics education; two-day festival featuring contributions of high school and college physics teachers; and at least one session of contributed papers on the use of micros in physics teaching.

For more information contact the American Association of Physics Teachers, Graduate Physics Building, SUNY at Stony Brook, Stony Brook, NY 11794; (516) 246-6840.

A general interest computer club has started in Evansville, IN. For information contact Robert Heerdink, Evansville Computer Club, c/o National Sharedata Corp., P.O. Box 3895, Evansville, IN 47737; (812) 426-2725.

Never No Shakes in California

Hamburger lovers in California don't have to worry about waiting in long lines only to discover that their favorite fast food restaurant is out of buns, shakes or fries.

Carl's Junior is satisfying all of its customers with the help of minicomputer systems and point-of-sales devices at approximately 100 of its 232 hamburger outlets in California. Manufactured by HLx Systems, Anaheim, CA, the System is designed as a control for placing customer food orders and providing efficient communications between cashiers and cooks.

The system in each restaurant consists of a front end cashier terminal, a small remote printer in the kitchen, a processor and an FD200 microfloppy disk drive manufactured by the Pertec Division of Pertec Computer Corp., Chatsworth, CA.

"When a customer places an order at the counter, a cashier touches the appropriate boxes on a terminal keyboard containing all the menu items, cooking instructions and miscellaneous reporting keys," explained Ron Paullins, District Manager at Carl's Junior. "The terminal totals the sale and adds tax. Depositing the money in the cash drawer triggers the system to print the food order on the cook's printer in the kitchen.

"The system has increased our sales by speeding up the ordering process," Paullins said. "Each store sells approximately 1000 hamburgers, 700 orders of french fries and 1500 soft drinks to about 800 customers per day. And each sale using a mechanical cash register used to take at least one minute to process. Now the ordering process takes about 28 seconds."

Paullins pointed out that aside from on-site improvements, the computer system has aided management in sales and inventory analysis. "Now, each time a cashier enters a sale," said Paullins, " it automatically enters the computer, where it's stored by the disk drive. By using inventory and sales data, the system can determine what items should be left over, giving managers a more precise measurement to use when ordering supplies."

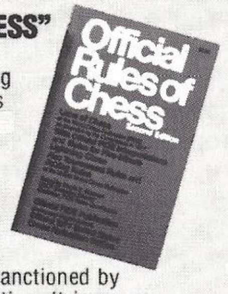
Since the system is programmed with the latest costs for every needed commodity, it can also figure the price for every item on the menu to determine production costs and profit margins, according to Paullins.

Because the system also keeps complete employee files with names and addresses, rates of pay, deductions for meals and hours worked, the computer can figure wages too.

"Presently, each store still has to call headquarters to order supplies. But within a year or two, when we've expanded, headquarters will be able to determine the amount of items each store needs based on analysis of data, and will order all supplies for all of the outlets," Paullins concluded.

Improve your chess with the right book for your level of skill

PLUS A FREE COPY OF "THE OFFICIAL RULES OF CHESS"

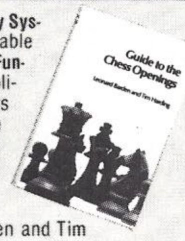


Learn from the masters. The David McKay Company, the leading publisher of chess books, has carefully selected the classic chess books for all levels of play as a special offer.
Each book is written by a chess master in an easy-to-understand style so that you can quickly improve your own game. And with your selection you'll receive "The Official Rules of Chess"—a $2.95 value—as a free gift. Order the books you need now and add years of greater enjoyment to your chess game.

The Official Rules of Chess. The only book of chess rules officially sanctioned by the World Chess Federation and the United States Chess Federation. It is yours to keep as **a free gift** just for examining any of the books offered here. **A $2.95 value—Free.**

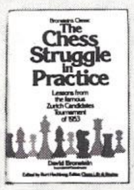
FOR BEGINNERS

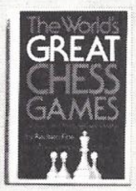
Chess Fundamentals by J.R. Capablanca and **My System** by Aron Nimzovich. The two most valuable books for anyone interested in chess. **Chess Fundamentals** is the classic manual that has simplified learning chess for thousands through its diagrams and explanations of 18 complete games. **My System** reveals the theories of positional chess essential to every player who wants to improve his or her game. A two-book set (paperbound) **Only $8.90**



Guide to the Chess Openings by Leonard Barden and Tim Harding. A basic, easy-to-use survey of the concepts behind the openings. Designed to help the developing player find his own way to stronger play. **Only $8.95**

FOR ADVANCED PLAYERS





Modern Chess Openings by Walter Korn. Eleventh Edition. This classic work is known as "The Chess Player's Bible." It is designed to give you a comprehensive view of the historic range and the structure of the openings, the essence of opening strategy, a summary of the important variations, and all the latest developments in the game. A "must" for any serious player. **Only $12.50**

The Modern Chess Sacrifice by Leonid Shamkovich. The first book to give you a detailed classification of sacrifices in the opening and middle games, with examples from contemporary Master play. Plus the first modern treatise on the theory of sacrifice. A valuable addition to your chess library. **Only $9.95**

The Chess Struggle in Practice by David Bronstein. The classic analysis of the historic Zurich Candidates Tournament which demonstrated some of the strongest and most innovative middle-game play in modern times. A fascinating study, complete with 350 game diagrams. **Only $17.95**

The World's Great Chess Games by Reuben Fine. A new, revised and expanded edition. A comprehensive coverage of the game from its historical origins to the beginnings of modern chess and to some of today's greatest players. An exceptional opportunity to learn from the greatest players in history. **Only $14.95**

Your Satisfaction Guaranteed Or Your Money Back

THE DAVID McKAY CO. INC. 750 Third Avenue, New York, N.Y. 10017 Dept. #741

YES. I wish to improve my chess game by learning from the Masters. Please send me the book(s) I have indicated below. If I am not completely satisfied, I can return it (them) within 30 days for a full refund or credit. I understand that with my order I **will also receive a free copy of The Official Rules of Chess** a $2.95 value, which is mine to keep just for examining the book(s) I have selected.

- ☐ Chess Fundamentals and My System, a two-book set @ $8.90
- ☐ Guide to the Chess Openings @ $8.95
- ☐ Modern Chess Openings @ $12.50
- ☐ The Modern Chess Sacrifice @ $9.95
- ☐ The Chess Struggle in Practice @ $17.95
- ☐ The World's Great Chess Games @ $14.95

☐ Enclosed is my check ☐ money order ☐ for $________
N.Y. State and Calif. residents, please add sales tax
☐ I prefer to charge my purchase to my
☐ Master Charge ☐ Visa ☐ American Express

Master Charge Bank Number ____________

Account Number ____________ Exp. Date ________

Signature ______________________

Please Print

Name ______________________

Address ______________________

City ____________ State ________ Zip ________

SPECIAL OFFER

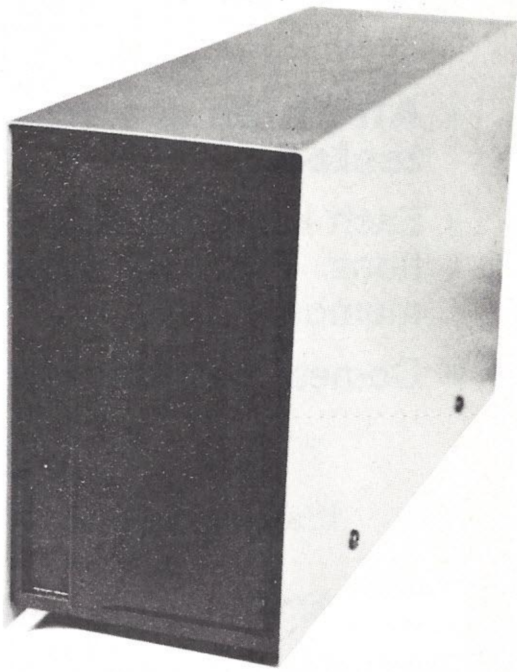
NEW RELEASE!!

Disk Drives For The TRS-80*

- Complete with 4 Drive Daisy Chain Cable
- Nothing else to buy. Complete with power supply. Just plug into your expansion interface.
- Fully assembled, tested in an attractive metal enclosure.
- Can be daisy chained up to 4 units.
- Full 1 year warranty.
- Complete manual and easy hookup. Just plug it in.

ONLY

$390^{00}



Requires 16K RAM, Level II Basic and Radio Shack TRS-80 expansion interface or World Power Systems expandor interface or disc controller module. See our catalog sheet. TRS-80 is a trademark of the Tandy Corp.

*TRS-80 is a trademark of Tandy Corp.

WORLD POWER SYSTEMS, INC.

1161 N. El Dorado Place, Suite 333, Tucson, Arizona 85715
24 Hour Order Phone No: 602-886-2537

CIRCLE 12

POWERFUL INTERFACES

TRS-80* SERIAL PARALLEL I/O MODULE

8-SERIAL INPUT/OUTPUT PORTS:
8-PARALLEL INPUT/OUTPUT PORTS:
MODEL MSIO-K $129.95
MODEL MSIO-A (ASSEM.) $149.95

- **EASY CONNECTION:** Connects to the expansion port edge card connector between keyboard and expansion interface or direct to rear of the TRS-80* keyboard.
- **DIP SWITCH:** All ports, baud rate, parity, etc. all set by dip switches.
- **ON BOARD FIRMWARE:** No software driver routine needed for operation of the module. Simple OUT and IN statements operate the module.
- **RS-232, CURRENT LOOP:** All 8 channels can be selected for RS-232 or current loop.
- **BAUD RATE SELECTION:** All channels dip switch selectable for individual baud rates from 110 to 9600 baud.

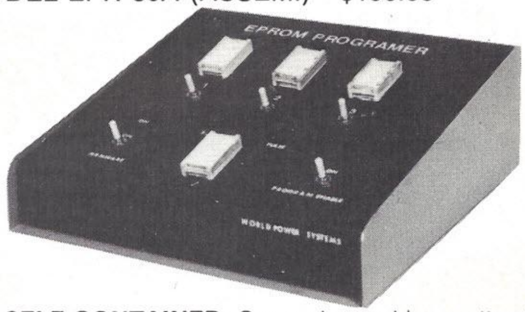
TRS-80* TO S-100 BUS

MODEL RSB-K (KIT) - $249.95
MODEL RSB-A (ASSEM.) - $289.95

- **FULLY SELF CONTAINED POWER SUPPLY.** (10 AMP).
- **BUS TERMINATION:** BUS termination and conditioning for no cross talk or noise etc.
- **S-100 SIGNALS:** All required S-100 signals are generated by on board logic and is fully compatible with the TRS-80*.
- **COMPLETE:** Comes complete with cabinet, card guides, on-off switch and sockets. Nothing else to buy.
- **STAND ALONE:** This system can stand alone or can operate with the TRS-80*. All input, output, address and signal lines fully buffered between TRS-80* and S-100 BUS system.
- **EASY CONNECTION:** Just plug it into the rear of the keyboard or between the keyboard and expansion interface. Also includes 40 pin edge connectors for connection to other interfaces.

TRS-80* EPROM PROGRAMMER +3

MODEL EPR-80K (KIT) - $129.95
MODEL EPR-80A (ASSEM.) - $159.95



- **SELF CONTAINED:** Comes housed in an attractive cabinet with self contained power supply.
- **PROGRAMS:** This unit programs the popular 2708, 2716 EPROMS. Personality modules for other EPROMS will be available at a later date.
- **FIRMWARE:** On board firmware so that no software need be written or entered into your CPU system. The firmware can be shut off when not in use. Firmware residents at F000. The firmware in and out of system is controlled from a switch on front panel.
- **3-ADDRESSABLE ROM LOCATIONS:** The EPROM Programmer has three sockets on front panel which are addressable to any location by dip switch. In addition each ROM location can be shut off or turned on by switches located on the front panel.
- **MONITOR:** A monitor is supplied within the firmware for performing several functions. Move memory, debug, verify, program from memory, program from TTY input, etc.
- **EASY CONNECTION:** The EPROM Programmer is attached with ease. For the TRS-80* users, the unit plugs into the rear of the keyboard or between the keyboard and expansion interface,. Included with the unit are two additional 40 pin edge connections for interfacing of other interfaces. For the S-100 users, a molded connection cable is supplied and it is inserted into one of the connectors on your mother board. Plug it in and it is ready to use.
- **FULLY BUFFERED:** add address and data lines are fully buffered.
- **OTHER FEATURES:** Other features include status lights for which ROM selected, switch enable for programming, pulse (burn) indicator firmware select-deselect switch, on, off and dip switches for the addressing of each ROM location.

*TRS-80 is a trademark of Tandy Corp.

WORLD POWER SYSTEMS, INC.
1161 N. El Dorado Place, Suite 333, Tucson, Arizona 85715
24 Hour Order Phone No: 602-886-2537

CIRCLE 16

POWERFUL INTERFACES

TRS-80* EXPANDOR INTERFACE

MODEL EI-80K (KIT) - $329.00
MODEL EI-80A (ASSEM.) - $349.00

- 32K high speed 250NS memory.
- Disc controller which controls mini or 8" floppies.
- RS-232 Port.
- Parallel Port
- Self contained heavy duty power supply. Plugs directly into rear of TRS-80* keyboard. Comes in attractive cabinet. **Twice the value for what you would spend for a TRS-80* expansion interface.**

TRS-80* MASTER CONTROL CONSOLE

MODEL MCC-K (KIT) - $129.95
MODEL MCC-A (ASSEM.) - $159.95

A COMPLETE COMMAND CENTER FROM YOUR KEYBOARD OR FROM ANY LEVEL II OR DISC BASIC PROGRAM. Turn on bells, sprinklers, sense fire and burglar alarm, anything that needs a switch can be controlled by the command center.

- **16 OUTPUT LINES:** With 8 relays, SPST, and 8TTL diode protected signals.
- **16 INPUT LINES:** 8 lines with OPTO-COUPLERS and 8 ITL diode protected.
- **FULL LED PANEL:** For status indicators of all control lines.
- **COMPLETE WITH CABINET:** Has attractive sloping cabinet.
- **FULLY HEAVY DUTY POWER SUPPLY:** Contains power supply. No external power needed.
- **EASY CONNECTION:** Plugs into TRS-80* expansion port edge card rear of keyboard or between keyboard and expansion interface.
- **2-EDGE CONNECTORS:** 2-additonal expansion 40 pin edge connectors.
- **NEEDS NO SOFTWARE:** Operates from OUT and IN statements from BASIC or machine code statements. Example: (Out 5, 1=turn on switch 5. Out 6, 1=turn off switch 5, etc.)
- **COMPLETE MANUAL AND SAMPLE PROGRAMS:** Comes with comprehensive manual and sample programs.

*TRS-80 is a trademark of Tandy Corp.

S-100 BUS MASTER CONTROL CARD

MODEL MCC-100K (KIT) - $159.95
MODEL MCC-100A (ASSEM.) - $189.95

TURN IT ON.....TURN IT OFF

Now you can control the outside world plus sense its status and its functions. 16 output and input lines. Turn on those bells, activate burglar alarms, etc.

- **16 OUTPUT AND INPUT CHANNELS:** 16 output channels with SPST relay on each. Opto-couplers on each one of the input channels.
- **EASY PORT ASSIGNMENT:** Port assignment is made via DIP SWITCH. In addition this board features our "ALL HARDWARE" software match setting features. You are able to select and set status, its parity to match any software configuration. No need to change the software to match the board.
- **SIMPLE OPERATION:** Turning off the relays is commandable by addressing a port, plus turning a bit on or off. Sample: You're in basic and you want to turn on switch 16. You would write out 3, 16. This turns on switch 16. To turn it off you would write out 4, 16 and off it is.
- **HIGH QUALITY:** The highest quality parts are used. The P.C. board is double sided with plated through holes, solder mask and silk screened legend.
- **FULL DOCUMENTATION:** A complete manual of operation and assembly is included.

TRS-80* DISC CONTROLLER MODULE

MODEL DCM-80K (KIT) - $159.95
MODEL DCM-80A (ASSEM.) - $189.95
Option available: 1) 16K RAM Kit, high speed 250 NS with purchase of board - Special $85.00.

- Has provisions for 16K memory.
- Will control mini or 8" floppies.
- DOS operating system included.
- Plugs directly into rear of TRS-80* keyboard.
- Complete with power supply in attractive cabinet.

S-100 8K STATIC 250NS RAM MEMORY CARD

MODEL 8K-100K (KIT) - $119.95
MODEL 8K-100A (ASSEM.) - $139.95

- Fully buffered address, control and data lines.
- Memory protect and unprotect.
- Power on clear.
- Bank select feature for selection to any 64K quadrant.
- Battery backup.
- Will run with any Z-80 Microprocessor without need of wait states.
- S-100 BUS power requirement 1.4 amps.

WORLD POWER SYSTEMS, INC.
1161 N. El Dorado Place, Suite 333, Tucson, Arizona 85715
24 Hour Order Phone No: 602-886-2537

CIRCLE 17

The Thoughts Behind the Structure

Your business needs a software package developed for your specific needs. The step-by-step procedure explained in this article will help you create a program fulfilling your particular business's requirements.

BY ROBERT T. NICHOLSON

In recent years, structured programming has been advanced as a cure-all for programming ills. Amateur and professional programmers alike have taken up stepwise refinement and Warnier-Orr diagrams, only to find that the resulting programs were often no better than their previous efforts.

There are, of course, explanations for these problems. To begin with, any new technique requires a certain amount of practice before it will yield results. Unfortunately, after the great claims made by many structured programming proponents, results are often expected to be immediate and miraculous. But a more significant reason for the failure of structured programming techniques is that they're viewed as a development "cookbook", rather than as tools for implementing well-planned software packages.

The important point here is that, for any set of tools to work, their user must have a good overall picture of what's being built. In programming, this overall picture (or plan) should answer these questions:

1. What is the purpose of the program?
2. Who will use it?
3. How will it be used?
4. What will it output?
5. What will it require as input?
6. What information must it store and process?

The resulting program plan can easily be filled in with a bit of brainstorming. Think about the difficulties likely to be encountered, the algorithms required and the special features desired. If a large amount of information is to be stored by the program, note any relationships among the data items which may help in deciding their internal representation. The structure of these notes is not important; the goal is simply to gather information and ideas for later use.

Such an overall view of the program requirements will not only provide a starting point for the stepwise refinement process, but will form a useful model or standard against which the program may be checked at each stage of its design and implementation.

To better understand the thought processes required to derive a good program plan, let's take a look at an example. Suppose that we would like a program to draw a bar-chart comparing a number of items (this is the sort of chart encyclopedias use to compare items such as grain production in developing nations). The first step is to develop the program plan by answering the questions previously listed:

1) The purpose of the program is to provide a visual comparison of a number of quantities (ie, a bar chart), with a minimum amount of work on the part of the user.
2) The author and a number of friends and co-workers will use the program. Users will be expected to know how to use the computer system itself, but the charting program should provide instructions and help as needed.
3) The program will be operated interactively by a single user at a time. Speed and efficiency will not be major considerations.
4) The program will produce a bar chart, where each bar is made up of several rows of Xs. Bars should be labelled, and a scale and title printed.
5) The user must enter the quantities to be shown, labels for the quantities and a scale and title for the chart.
6) The program must be capable of saving all of the data entered by the user until it is ready to print the chart. Processing will be limited to dividing the data into suitable size intervals to fit across the page.

The above information provides a pretty clear picture of what will be needed, but it can still be improved with more thought. The following are a few ideas which might arise after a bit of brainstorming:

- The number of quantities which can be saved and charted will be limited only by the available array storage space. It would be nice if the user could enter fewer than the maximum number of values. We will therefore need a special way of saying "end of data". We will use a double colon (::) for this purpose.
- For the sake of simplicity, we will plan on running the bars across the page, with their base near the left edge. Also, we will restrict the user to com-

paring positive quantities only.

- Requiring the user to enter the scale seems like an unnecessary inconvenience. We can actually compute the scale by taking the largest value entered by the user, padding it just a bit to make the chart less crowded, and then dividing it to fit into the available space. We can then give the user a choice of using this computed scale or entering a different one.
- Since a number of people will use this program, it should be as foolproof as possible. Information and responses entered by the user should be checked for legality and correctness.

These decisions are, of course, based on one program and the imagined needs of its author; a different set of goals and ideas will occur for each program planned. Whatever the final plan includes, it will form the basis for the structured programming process. For a good initial breakdown for our bar-chart program, see Figure 1.

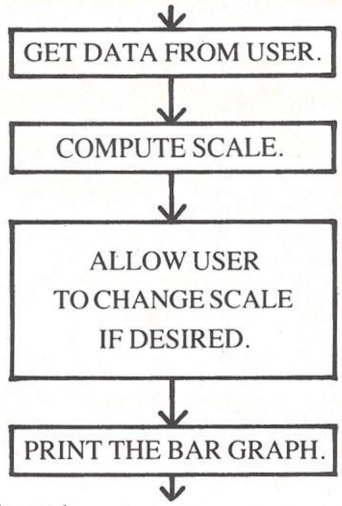


Figure 1

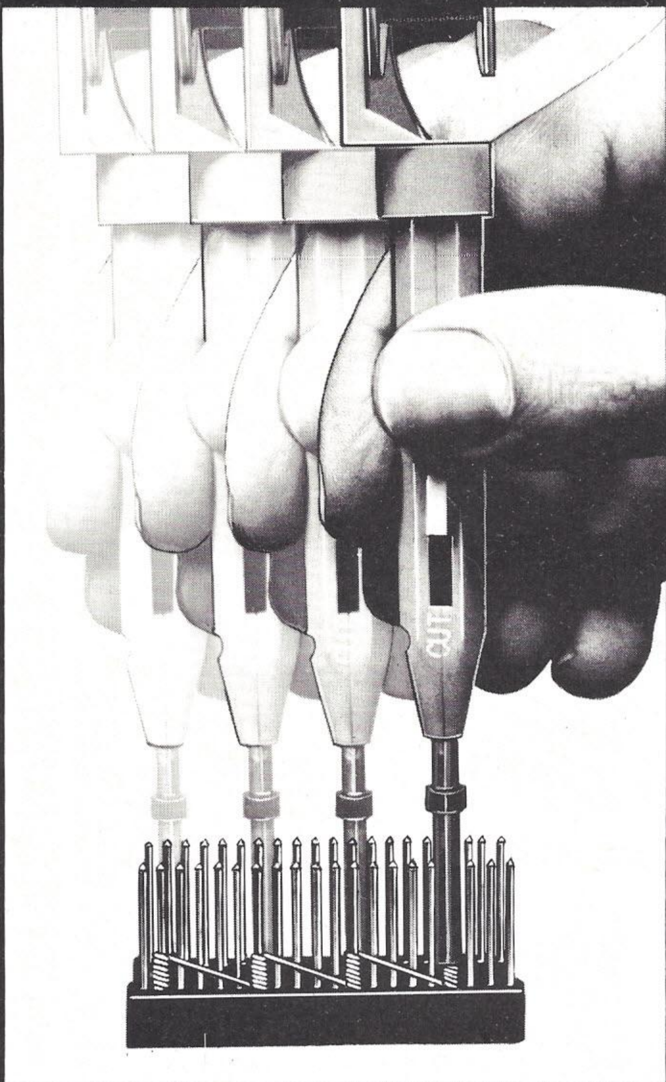
Using stepwise refinement, the program can be further transformed into a pseudo-code (or design language) representation. After translating selection and iteration structures into equivalent BASIC statements, we might end up with the program shown in the Program Listing. Note that the initial and second-level refinements are shown as remarks in the BASIC code. A sample run of the program, following the listing, demonstrates that it meets all of the goals set in the program plan: the user can enter a variable number of data values, label the values and choose to use a computed scale or enter a new scale. Errors or invalid responses to questions are caught whenever possible. The program has been well tested to protect its users against unpleasant surprises.

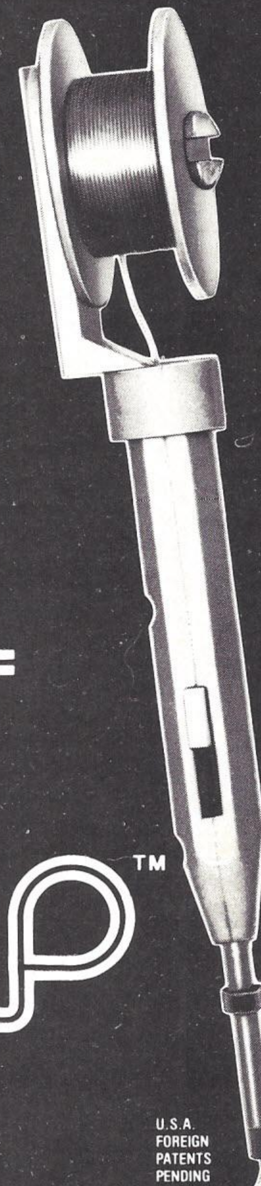
Although this program was quite simple to develop, it was made considerably more valuable to its author by some careful thought before beginning the design; large and complex projects can be greatly enhanced through the use of program planning techniques. So, before you sit down to write your next structured program, do a little planning; you may find that those promised benefits can be yours after all. □

Program Listing

```
 10 REM ****************************************************************
 20 REM
 30 REM  BAR  - Prints a bar chart of up to 20 positive values
 40 REM         entered by the user. Allows the user to accept
 50 REM         an automatically generated scale,  or  enter a
 60 REM         scale manually.
 70 REM
 80 REM ****************************************************************
 90 REM
100 REM  VARIABLE USAGE:
110 REM
120 REM  A$     ANSWER     String variable for user's answers to question
130 REM  B      BASE       Starting value for bar graph.
140 REM  C      COUNT      The number of values entered by the user.
150 REM  H$     HEADER     The graph header input by the user.
160 REM  L$     LABELS     An array of graph labels input by the user.
170 REM  I      INDEX      Temporary index variable.
180 REM  I1     INCR1      Increment between columns of graph.
190 REM  I2     INCR2      Increment between every 10 columns (labels).
200 REM  J      JTEMP      Temporary storage.
210 REM  K      KTEMP      Temporary storage.
220 REM  M      MAX        Largest value entered by the user.
230 REM  M1     MULT       The multiplier used in computing magnitude.
240 REM  M2     MAG        The magnitude of MAX.
250 REM  R      ROUND      The upward-rounded value of MAX.
260 REM  V      VALUES     An array of values input by the user.
270 REM
280 REM ****************************************************************
290 DIM L$[20,10],A$[1],H$[72]
300 DIM V[20]
310 REM
320 REM  GET DATA FROM USER:
330 REM
340 REM      PRINT INSTRUCTIONS TO USER:
350 REM
360 PRINT
370 PRINT "BAR  will print a bar chart of up  to  twenty  positive"
380 PRINT "values entered by the user.  For each bar, you will  be"
390 PRINT "prompted for a label, and a value.  Labels may be up to"
400 PRINT "ten characters in length; values must not be  less than"
410 PRINT "zero.  If you wish to print fewer than twenty bars, you"
420 PRINT "may terminate the input phase by typing two colons (::)"
```

NEW!





WHY CUT?
WHY STRIP?
WHY SLIT?

WHY NOT...

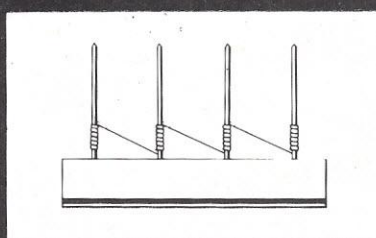
JUST WRAP™

U.S.A.
FOREIGN
PATENTS
PENDING

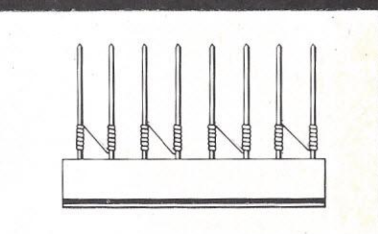
- AWG 30 Wire
- .025" Square Posts
- Daisy Chain or Point To Point
- No Stripping or Slitting Required
  ....JUST WRAP TM....
- Built In Cut Off
- Easy Loading of Wire
- Available Wire Colors:
  Blue, White, Red & Yellow

***$14.95**

JUST WRAP TOOL WITH ONE 50 FT. ROLL OF WIRE		
COLOR	PART NO.	U.S. LIST PRICE
BLUE	JW-1-B	$14.95
WHITE	JW-1-W	14.95
YELLOW	JW-1-Y	14.95
RED	JW-1-R	14.95
REPLACEMENT ROLL OF WIRE 50 FT.		
BLUE	R-JW-B	2.98
WHITE	R-JW-W	2.98
YELLOW	R-JW-Y	2.98
RED	R-JW-R	2.98



DAISY CHAIN



POINT TO POINT

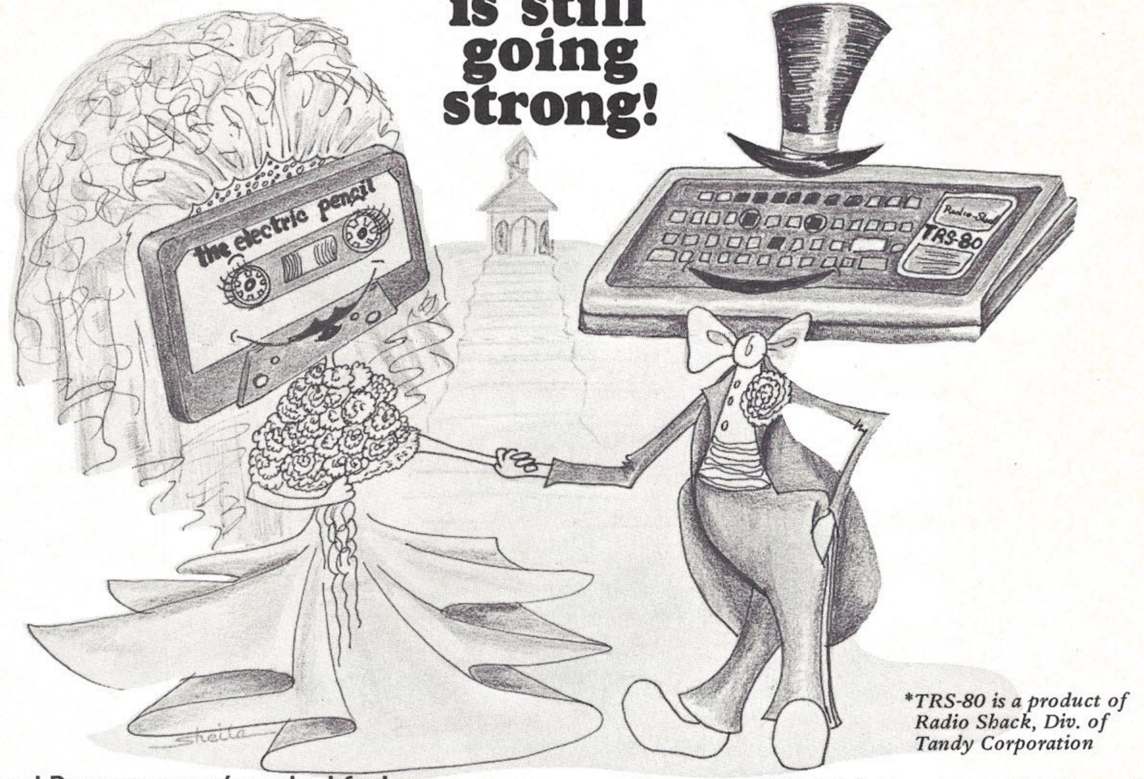
ok® MACHINE & TOOL CORPORATION 3455 CONNER ST., BRONX, N.Y. 10475 (212) 994-6600/TELEX 125091

*MINIMUM BILLING $25.00 / AD SHIPPING CHARGE $2.00 / NEW YORK CITY / STATE RESIDENTS ADD APPLICABLE TAX.

```
430 PRINT "when prompted for the label."
440 PRINT
450 REM
460 REM       INPUT ITEM LABELS AND VALUES UNTIL 20 HAVE BEEN
470 REM       ENTERED, OR THE USER ENTERS A DOUBLE COLON (::).
480 REM
490 LET C=1
500 IF C>20 OR L$[C]="::" THEN 620
510 PRINT
520 PRINT "ENTER LABEL FOR BAR";C;":";
530 INPUT L$[C]
540 IF L$[C]="::" THEN 610
550 PRINT "ENTER VALUE FOR BAR";C;":";
560 INPUT V[C]
570 IF V[C]>=0 THEN 600
580 PRINT "VALUES MUST NOT BE LESS THAN ZERO."
590 GOTO 550
600 LET C=C+1
610 GOTO 500
620 LET C=C-1
630 REM
640 REM       INPUT HEADER FOR GRAPH:
641 REM
650 PRINT
660 PRINT "ENTER A HEADER FOR THE GRAPH:"
670 INPUT H$
680 PRINT
690 REM
700 REM  COMPUTE SCALE:
710 REM
720 REM       SET BASE VALUE FOR BAR CHART:
730 REM
740 LET B=0
750 REM
760 REM       FIND MAXIMUM OF VALUES ENTERED BY THE USER:
770 REM
780 LET M=0
790 FOR I=1 TO C
800   IF V[I]<=M THEN 820
810   LET M=V[I]
820 NEXT I
830 REM
840 REM       ROUND UP MAX. VALUE TO DETERMINE SCALE OF GRAPH:
850 REM
851 REM            DETERMINE MULTIPLIER TO USE IN ESTIMATING
852 REM            MAGNITUDE OF MAX VALUE.  USE .1 IF MAX IS
853 REM            LESS THAN 1, OR 10 IF MAX IS GREATER THAN 1:
854 REM
860 LET R=M
870 LET M1=10
880 LET M2=1
890 IF R>1 THEN 910
900 LET M1=.1
901 REM
902 REM            DETERMINE MAGNITUDE OF MAX VALUE BY SUCCESSIVE
903 REM            DIVISIONS BY THE MULTIPLIER.
904 REM
910 IF R<10 AND R>=1 THEN 950
920 LET M2=M2*M1
930 LET R=M/M2
940 GOTO 910
941 REM
942 REM            ROUND MAX VALUE UPWARD BY ADDING 1 AND TRUNCATING,
943 REM            THEN RESTORE TO PROPER MAGNITUDE:
944 REM
950 LET R=INT(R+1)*M2
960 REM
970 REM       DIVIDE ROUNDED VALUE BY NUMBER OF COLUMNS (50)
980 REM       AND NUMBER OF LABELLING INCREMENTS (5) TO PRODUCE SCALE:
990 REM
1000 LET I1=R/50
1010 LET I2=R/5
1020 REM
```

```
1030 REM  ALLOW USER TO CHANGE SCALE IF DESIRED:
1040 REM
1050 REM      DISPLAY CURRENT SCALE  (BASE AND INCREMENT):
1060 REM
1070 PRINT
1080 PRINT "The bar graph will now start at";B;"and proceed to";R
1090 PRINT "in five increments of";I2;"."
1100 PRINT
1110 REM
1120 REM      ASK USER IF SCALE IS TO BE CHANGED:
1130 REM
1140 PRINT "DO YOU WISH TO CHANGE THE BASE AND/OR INCREMENT?  (Y or N)";
1150 INPUT A$
1160 IF A$="N" THEN 1360
1170 IF A$="Y" THEN 1240
1180 PRINT "PLEASE RESPOND WITH A 'Y' OR AN 'N'"
1190 GOTO 1100
1200 REM
1210 REM      IF SCALE IS TO BE CHANGED, ACCEPT NEW SCALE  (BASE
1220 REM      AND INCREMENT) FROM USER:
1230 REM
1240 PRINT "ENTER NEW BASE:";
1250 INPUT B
1260 PRINT "ENTER NEW INCREMENT:";
1270 INPUT I2
1280 LET I1=I2/10
1290 IF (B+5*I2)>M THEN 1360
1300 PRINT
1310 PRINT "WITH THE NEW BASE AND INCREMENT, THE MAX VALUE (";M;")"
1320 PRINT "WILL NOT FIT ON THE GRAPH.  THE BASE PLUS 5 TIMES THE"
1330 PRINT "INCREMENT MUST BE GREATER THAN THE MAX VALUE."
1340 PRINT
1350 GOTO 1240
1360 REM
1370 REM  PRINT THE BAR GRAPH:
1380 REM
1390 REM      PRINT HEADER:
1400 REM
1410 PRINT
1420 PRINT
1430 PRINT H$
1440 PRINT
1450 PRINT
1460 REM
1470 REM      PRINT SCALE FOR BAR GRAPH:
1480 REM
1490 PRINT TAB(10);B;TAB(30);B+2*I2;TAB(50);B+4*I2
1500 PRINT TAB(20);B+I2;TAB(40);B+3*I2;TAB(60);B+5*I2
1510 PRINT TAB(12);"'
1520 REM
1530 REM      FOR EACH DATA ITEM, PRINT LABEL, VALUE, AND BAR:
1540 REM
1550 FOR I=1 TO C
1560   PRINT L$[I];TAB(12);
1570   LET J=(V[I]-B)/I1
1580   LET J=INT(J+.5)
1590   FOR K=1 TO J
1600     PRINT "*";
1610   NEXT K
1620   PRINT "*"
1630   PRINT V[I];TAB(12);
1640   FOR K=1 TO J
1650     PRINT "*";
1660   NEXT K
1670   PRINT "*"
1680   PRINT TAB(12);
1690   FOR K=1 TO J
1700     PRINT "*";
1710   NEXT K
1720   PRINT "*"
1730   PRINT
1740 NEXT I
1750 END
```

the marriage of the electric pencil™ to TRS-80* is still going strong!



TRS-80 is a product of Radio Shack, Div. of Tandy Corporation

TRS-80 disk version now available only $150.

NOW! The Word Processor you've asked for!

The well known Electric Pencil, which has been in use for over two years with most popular microcomputers, has also been available for use on the TRS-80. We are now proud to announce immediate availability of The Electric Pencil Disk Version for the TRS-80, featuring the same fine capabilities that have made the Electric Pencil so popular. Full price for the TRS-80 version is only $100, and for the TRS-80 disk version, only $150.

the electric pencil

—a proven word processing system

Extremely easy to use, The Electric Pencil lets you produce mailing lists, business forms, large numbers of original correspondence, camera-ready copy for printing . . . all on your TRS-80.

The Electric Pencil is a character oriented word processing system, which means you have maximum freedom and simplicity in the handling of text. No carriage returns or word hyphenations are necessary. Line formatting is done automatically by The Electric Pencil.

The Electric Pencil combines features found in more expensive systems with many capabilities you won't find anywhere. Right margin justification, page numbering and titling, and many combinations of line length, page length, and line and page spacing are always under your control. Text editing capabilities eliminate guesswork when inputting. The text appears on the video monitor as you insert it, and you can view text whenever you wish using the bidirectional variable speed scrolling feature. You can insert, delete, or relocate any text using simple keyboard commands.

The Electric Pencil has been designed to work with both Level I (16K system) and Level II models of the TRS-80, and with virtually any printer you choose. Level II owners will want the TRS-80 disk version, which is supplied on cassette. Transfer of the Electric Pencil to your own disk is as simple as entering a command.

Upgraded Versions

The Electric Pencil TRS-80 Disk Version is fully interactive with the DISK READ, WRITE, DIR, and KILL routines of TRSDOS 2.1. Any version of The Electric Pencil may be upgraded to accomodate new equipment. For instance, if you are already using The Electric Pencil and have added a floppy disk to your TRS-80 system, simply return the original cassette to Michael Shrayer Software, along with the price difference between your present Electric Pencil and the upgraded version, plus $15 for handling.

Demand a Demo from your Dealer!

Your dealer will have a manual and descriptive literature for you to see, and can demonstrate The Electric Pencil to you, on one or more of his demonstration microcomputers. Look the manual over carefully, and note the explicit instructions which lead you easily through The Electric Pencil operation. The manual was produced using The Electric Pencil which will enable you to see its many capabilities for yourself. Then try it out on your dealer's demo unit . . . most any microcomputer will do. If, for some incredible reason, he's unprepared, demand it! or write:

mss

MICHAEL SHRAYER SOFTWARE, INC.
1253 Vista Superba Drive
Glendale, CA 91205

CIRCLE 20

Program Run

```
BAR  will print a bar chart of up  to  twenty  positive
values entered by the user.  For each bar, you will  be
prompted for a label, and a value.  Labels may be up to
ten characters in length; values must not be  less than
zero.  If you wish to print fewer than twenty bars, you
may terminate the input phase by typing two colons (::)
when prompted for the label.


ENTER LABEL FOR BAR 1     :?SO. CAL.
ENTER VALUE FOR BAR 1     :?23500

ENTER LABEL FOR BAR 2     :?NO. CAL.
ENTER VALUE FOR BAR 2     :?19400

ENTER LABEL FOR BAR 3     :?OREGON
ENTER VALUE FOR BAR 3     :?11200

ENTER LABEL FOR BAR 4     :?WASH.
ENTER VALUE FOR BAR 4     :?13600

ENTER LABEL FOR BAR 5     :?NEVADA
ENTER VALUE FOR BAR 5     :?7200

ENTER LABEL FOR BAR 6     :?ARIZONA
ENTER VALUE FOR BAR 6     :?13700

ENTER LABEL FOR BAR 7     :?::

ENTER A HEADER FOR THE GRAPH:
?WESTERN REGION SALES  (MARCH):


The bar graph will now start at 0    and proceed to 30000
in five increments of 6000    .

DO YOU WISH TO CHANGE THE BASE AND/OR INCREMENT?  (Y or N)?Y
ENTER NEW BASE:?0
ENTER NEW INCREMENT:?4000

WITH THE NEW BASE AND INCREMENT, THE MAX VALUE ( 23500    )
WILL NOT FIT ON THE GRAPH.  THE BASE PLUS 5 TIMES THE
INCREMENT MUST BE GREATER THAN THE MAX VALUE.

ENTER NEW BASE:?0
ENTER NEW INCREMENT:?5000


WESTERN REGION SALES  (MARCH):


           0                   10000               20000
                     5000                15000               25000
           .         .         .         .         .         .
SO. CAL.   ***********************************************
 23500     ***********************************************
           ***********************************************

NO. CAL.   ***************************************
 19400     ***************************************
           ***************************************

OREGON     **********************
 11200     **********************
           **********************

WASH.      ***************************
 13600     ***************************
           ***************************

NEVADA     **************
 7200      **************
           **************

ARIZONA    ***************************
 13700     ***************************
           ***************************
```

Color. VP-590 add-on Color Board allows program control of 8 brilliant colors for graphics, color games. Plus 4 selectable background colors. Includes sockets for 2 auxiliary keypads (VP-580). $69.*

Sound. VP-595 Simple Sound Board provides 256 tone frequencies. Great for supplementing graphics with sound effects or music. Set tone and duration with easy instructions. $24.*

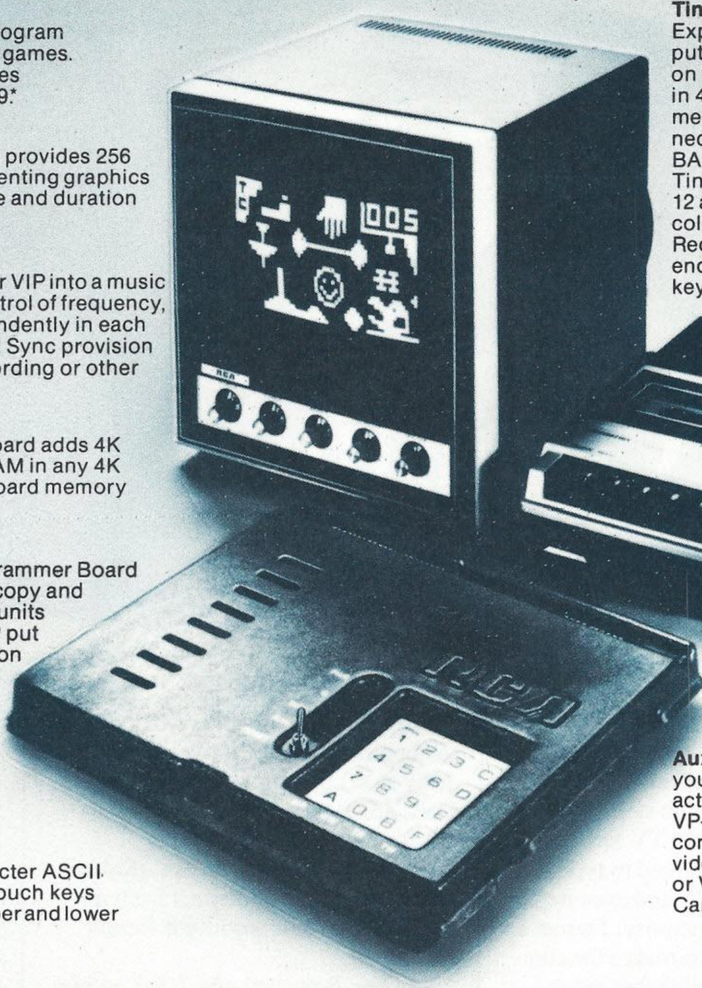
Music. VP-550 Super Sound Board turns your VIP into a music synthesizer. 2 sound channels. Program control of frequency, time and amplitude envelope (voice) independently in each channel. Program directly from sheet music! Sync provision for controlling multiple VIPs, multitrack recording or other synthesizers. $49.*

Memory. VP-570 RAM Expansion Board adds 4K bytes of memory. Jumper locates RAM in any 4K block of up to 32K of memory. On-board memory protect switch. $95.*

EPROM Programmer. VP-565 EPROM Programmer Board comes complete with software to program, copy and verify 5-volt 2716 EPROMs—comparable to units costing much more than the VP-565 and VIP put together! Programming voltages generated on board. ZIF PROM socket included. $99.*

EPROM Interface. VP-560 EPROM Interface Board locates two 5-volt 2716 EPROMs (4K bytes total) anywhere in 32K of memory. VIP RAM can be re-allocated. $34.*

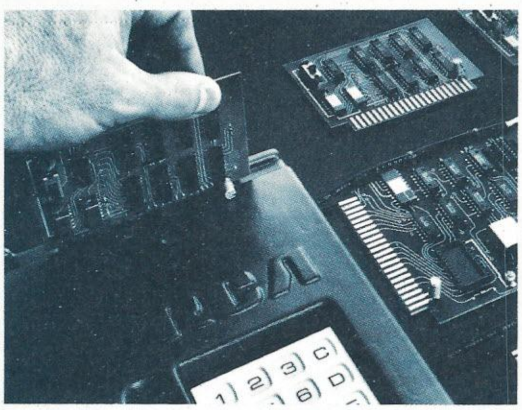
ASCII Keyboard.** Fully encoded, 128-character ASCII encoded alpha-numeric keyboard. 58 light touch keys including 2 user defined keys! Selectable upper and lower case. Handsomely styled. Under $50.*



Tiny BASIC.** VP-700 Expanded Tiny BASIC Board puts this high-level language on your VIP. BASIC stored in 4K of ROM. Ready for immediate use—no loading necessary. This expanded BASIC includes the standard Tiny BASIC commands plus 12 additional—including color and sound control! Requires external ASCII-encoded alpha-numeric keyboard. $39.*

Auxiliary Keypads. Program your VIP for 2-player interaction games! 16-key keypad VP-580 with cable ($15*) connects to sockets provided on VP-590 Color Board or VP 585 Keyboard Interface Card ($10*).

COSMAC VIP lets you add computer power a board at a time.



With these new easy-to-buy options, the versatile RCA COSMAC VIP (CDP18S711) means even more excitement. More challenges in graphics, games and control functions. For everyone, from youngster to serious hobbyist. And the basic VIP computer system starts at just $249* assembled and ready to operate.

Simple but powerful—not just a toy.

Built around an RCA COSMAC microprocessor, the VIP includes 2K of RAM. ROM monitor. Audio tone with a built-in speaker. Plus 8-bit input and 8-bit output port to interface relays, sensors or other peripherals. It's easy to program and operate. Powerful CHIP-8 interpretive language gets you into programming the first evening. Complete documentation provided.

Take the first step now.

Check your local computer store or electronics parts house. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

*Suggested retail price. CDP18S711 does not include video monitor or cassette recorder.
**Available 1st Quarter, 1979.

The fun way into computers.

RCA

CIRCLE 21

"G" is for Graphics

BY MARK ZIMMERMANN

I have a 2-1/2 year old neighbor named Ray. He's learning the alphabet, and gets a tremendous thrill out of pushing a key and seeing the same symbol appear, magically, on the TV screen of my PET. He's also learned to type in "RUN", hit "return", and execute programs loaded in memory. I wrote a program for him which makes the computer a personalized alphabet picture book. It lets an adult (or older child) define a picture for every key on the keyboard: A is for Apple, . . . , Z is for Zilog, and so forth. Pressing any key immediately produces that picture on the screen. Pictures are stored in data files: a completely new picture book can be loaded in minutes.

This program has a great educational potential. There's instantaneous positive feedback to the student as he learns the letters (and eventually, words). New drawings can be conveniently made if old ones are uninteresting. Most important of all, the child at an early age gets an introduction to a computer as an exciting friend which does what one commands, *not* as an incomprehensible, uncontrollable monster!

The program to accept, store and produce these pictures is written in BASIC, and should be adaptable to other manufacturers' machines without much trouble, if common string functions are available. First I'll discuss the general structure of the program, and then spend time on the special features necessitated by PET peculiarities.

All pictures are stored as strings. To each key on the keyboard corresponds a number, provided by the function ASC: for key A$, the number is ASC(A$). At the beginning of the program, a string array (a set of strings) is dimensioned, from Z$(0) through Z$(255). When any key is pushed, the string Z$(ASC(A$)) is printed out. (Shifted keys have symbols distinct from unshifted ones, and produce different strings.)

If a key is pushed which has no picture defined for it yet, (i.e., when a new picture book is being produced), the program branches to steps 1000 to 1280. These take in characters from the keyboard and store them, one by one, in one of the Z$(I) strings. A picture is built up out of special symbols and ordinary ASCII characters available, plus cursor control characters. Hitting the "return" key ends the picture definition; of course, any other key can be used for such an end marker, if desired, by changing the "13" in line 1220.

If the "clear screen" key ("clr") is pushed, the previously displayed picture is erased from memory (the previous key is stored in AP$) and the program goes into the picture-defining mode.

Finally, if the "home" key is pushed, the system goes to the tape reading or writing mode. If read is selected and a data file name is given, the tape is scanned for that file. When found, it is read into memory; old pictures are erased. If write mode is chosen, the present set of defined pictures is written to the cassette under a chosen file name. In either case, when the tape operation is finished, control returns to the first, keyboard-picture-production, part of the program.

Now, for some special features, applicable only to the Commodore PET machine: to begin with, a matter of notation. Items within quotes which are written lower-case in the listing refer to cursor-control characters. For example, in line 20, "6 down" means to insert six cursor-down commands there, within the quotation marks. The string C$ is a centering string; it moves the cursor to a convenient position for starting most pictures, and puts it here again when each picture is printed out.

The "GET A$" in line 100 gets a single character from the keyboard; if no key is pressed, it returns with A$="", an empty string. Line 10 loops until something is pressed, then execution continues.

Memory locations 224 and 225 contain the address of the beginning of the current line on the screen, in base-256. Specifically, the number 256*PEEK (225)+PEEK(224) gives the memory location of the leftmost character in the line where characters are being output. Location 226 gives the position in the line of the cursor, a number from 0 to 39. (I found these by peeking around memory while trying a variety of print operations.) Variable CL in the program thus is the current cursor location, and CH is the character there when line 1050 was executed. It is necessary to save this information so that the "delete" or "insert" or other cursor control keys print nicely while pictures are being defined, and don't simply write over whatever was in a given location. Of course, this information

may be very useful for other programming applications, too.

Line 1070 turns *on* the 2^7 bit of the current cursor location, which causes the character printed there to become "reverse-field". The logical operation "CH OR 128" means "turn *on* any bit which is either *on* in variable CH, or *on* in 128, or both". The time is then set, and lines 1100 to 1120 execute until TI, the internal timer, has increased by 15. Since each unit of TI is 1/60th of a second, this is 1/4 second elapsed time. After that, if no character on the keyboard had been pushed, the reverse-field feature is turned off by the logical operation "CH AND 127", which means "turn *on* any bit which is *on* in both variable CH and in 127". In binary, since 127 base 10 becomes "01111111", this simply turns *off* the leftmost, 2^7, bit of the cursor location, and makes the character there non-reverse-field. Lines 1160 to 1180 then loop for 1/4 second more, and if still no key has been pushed, the program goes back to line 1070, where the timer is reset and the character flashes into reverse-field again.

Line 1260 checks to see whether or not a cursor control character was pressed (ASCII 0 to 31 or 128 to 159). If it was a cursor control, the previous character CH is restored to location CL; an ordinary character is simply printed.

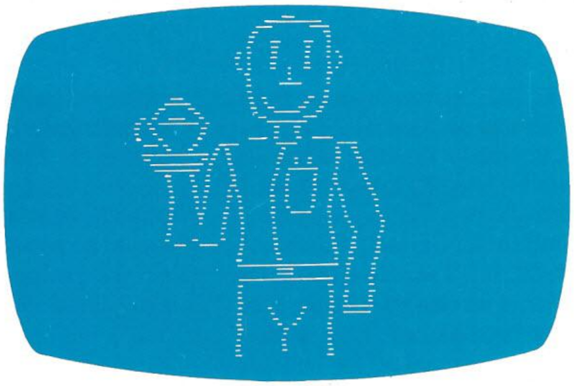
The PET's BASIC only allows strings of length 255 or less. So, line 1270 checks, and if the maximum length of string has been put in, that string is automatically ended. Without this line, adding the 256th character to a string causes a string error message to be printed, and the program stops. It would be possible to define larger strings, perhaps by putting characters above the 255th into another array Y$(I), but I have generally been able to make fine pictures within the 255 character limitation, and the extra feature would add to the program's complexity.

The only remaining peculiarity of the PET that appears in the program structure is in the way it writes data files. Data to be written to tape is accumulated until about 190 characters are ready for output, and then that block is written as a chunk onto the cassette. On early PETs not enough space was left on the tape between blocks of data out, and errors resulted when data was being read back in. Subroutine 5000 to 5040 gives a blip of current to the tape recorder motor (controlled by an interface at memory location 59411) to leave a gap on the tape between data blocks. For simplicity, I call that subroutine between every string that is saved. It actually only needs to be called every 190 characters, but keeping track of the number of characters is too much trouble; the method in the program listing works fine, and only costs a few centimeters of blank tape per file. The need for a gap between blocks of output is also the motivation behind lines 4030 to 4060, which split big pictures (length greater than 128) up into two chunks, A$ and B$ in lines 4050 to 4060.

Finally, and most peculiarly, my PET omits the first cursor control character that it prints to tape in each string. That's the sole reason for adding the "right" cursor to the beginning of each string. If your machine doesn't do that, the "right" additions can be omitted. As far as I know, it's just a part of the operating system, perhaps related to how strings are defined in memory — not a dangerous bug, as long as you know it's there and take care of it.

That's all there is to publishing your own picture book! Besides being wonderful for kids, the program makes a great demonstration for adult visitors. Artists, it's all yours! □

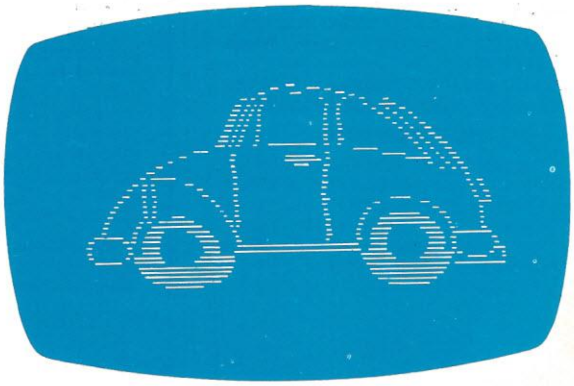
Program Listing



```
10   DIM Z$ (255)
20   C$="home, 6 down, 16 right"
50   PRINT "clr"
80   A$=""
100  GET A$: IF A$="" GOTO 100
120  IF A$="home" GOTO 2000: REM ENTER TAPE READ/WRITE ROUTINE
140  IF A$="clr" THEN A$=AP$: Z$(A)="": REM CLEAR PREVIOUS PICTURE
200 A=ASC(A$): AP$=A$: IF Z$(A)="" GOTO 1000: REM IF NO KNOWN PICTURE,
    GO TO PICTURE DEFINING ROUTINE
220  PRINT "clr";C$;Z$(A): GOTO 80
1000 PRINT "clr PLEASE DRAW ME SOMETHING FOR ";A$;C$;
1050 CL=256*PEEK(225)+PEEK(224)+PEEK(226):CH=PEEK(CL):
     B$=""
1060 REM CL IS CURSOR LOCATION, CH IS PREVIOUS CHARACTER THERE
1070 POKE CL,(CH OR 128):REM REVERSE FIELD AT CURSOR
1080 T=TI: REM SET TIMER
1100 GET B$: IF B$<>"" GOTO 1200: REM JUMP OUT IF CHARACTER INPUT
1120 IF (TI-T)< 15 GOTO 1100:REM KEEP LOOKING FOR CHARACTER FOR
     0.25 SECONDS
1140 POKE CL,(CH AND 127):REM TURN OFF REVERSE FIELD
1160 GET B$: IF B$<> "" GOTO 1200:REM JUMP OUT IF KEY IS HIT
1180 IF (TI-T) < 30 GOTO 1160: REM STAY .25 SEC MORE
1190 GOTO 1070:REM GO BACK AND DO IT AGAIN
1200 B=ASC(B$)
1220 IF B=13 GOTO 220: REM CARRIAGE RETURN ENDS PICTURE INPUT
1250 Z$(A)=Z$(A)+B$: PRINT B$;: REM ADD CHARACTER TO PICTURE
1260 IF (B < 32) OR ((B >127) AND (B <160)) THEN POKE CL,CH: REM
     CURSOR CONTROL CHAR'S DO NOT ERASE PREVIOUS CHARACTER
1270 IF (LEN(Z$(A)) >254) GOTO 220: REM TO PREVENT CRASH WHEN
     STRING GETS TOO LONG
```

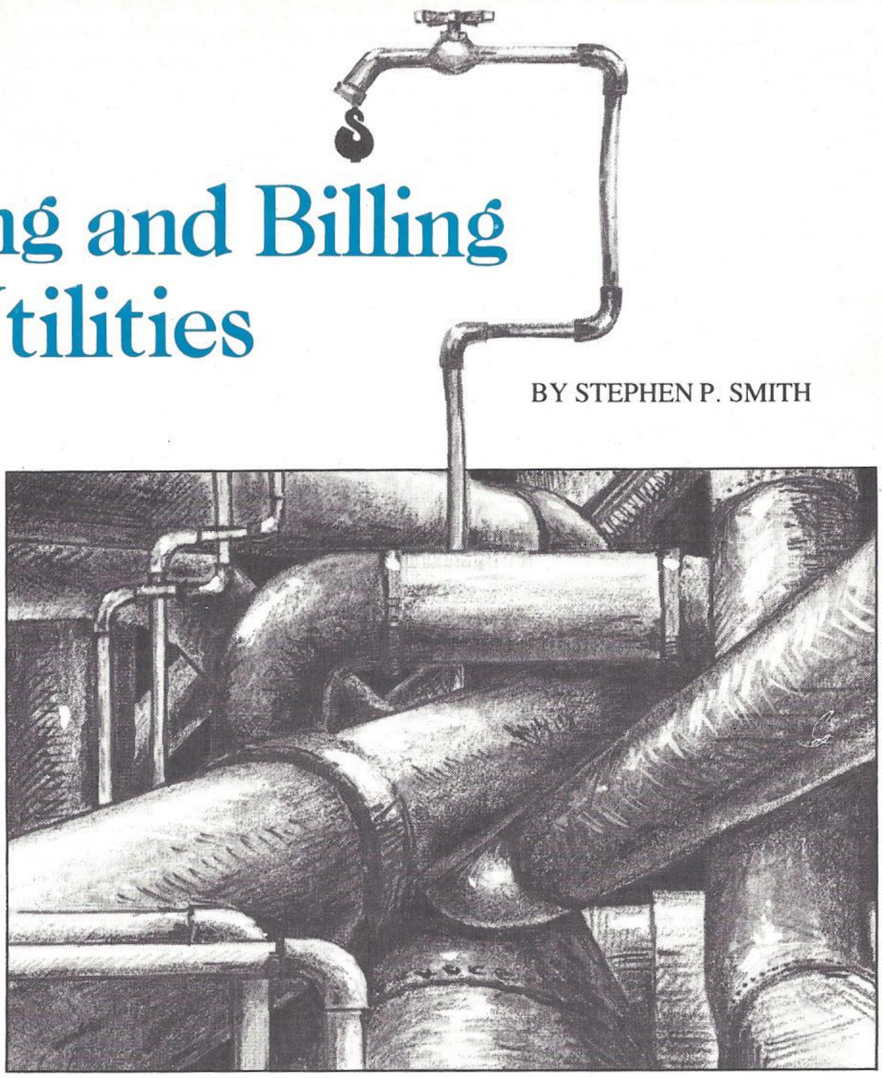
Program Listing Continued

```
1280 GOTO 1050: REM GO GET ANOTHER CHARACTER
2000 INPUT "READ OR WRITE";B$: B$=LEFT$(B$,1)
2020 IF B$="R" GOTO 3000:  REM GO TO READ TAPE ROUTINE
2040 IF B$="W" GOTO 4000: REM GO TO WRITE TAPE ROUTINE
2060 GOTO 2000
3000 INPUT "FILENAME";B$: OPEN 1,1,0,B$:INPUT #1,A$: IF A$<> B$ GOTO 3500:
     REM ABORT IF FILE HEADER DISAGREES
3010 PRINT "READING";A$
3020 FOR I=0 TO 255: Z$(I)="": NEXT I: REM CLEAR OUT OLD STRINGS
3030 INPUT #1,I: IF I=999 GOTO 3300: REM 999 MARKS END OF FILE
3040 PRINT CHR$(I): REM PRINT NAME OF PICTURE BEING READ IN
3050 GET #1,A$: IF ASC(A$)=13 GOTO 3030: REM CARRIAGE RETURN (=13)
     ENDS INPUT PICTURE
3060 Z$(I)=Z$(I)+A$: GOTO 3050: REM ADD CHARACTER TO PICTURE & CONTINUE
3300 CLOSE 1: PRINT "FILE READ":GOTO 80
3500 CLOSE 1: PRINT "PROBLEM--FILE ";B$;" HAS HEADER= ";A$;" FILE NOT
     READ"
3520 GOTO 80
4000 INPUT "FILENAME";B$: OPEN 1,1,1,B$: PRINT #1,B$
4020 FOR I=0 TO 255:. IF Z$="" GOTO 4100: REM SKIP EMPTY STRINGS
4030 PRINT #1,I: GOSUB 5000: L=LEN(Z$(I)): IF L< 129 GOTO 4080:
     REM SUBROUTINE 5000 SPACES TAPE FOR EARLY PETS
4040 REM SHORT PICTURES CAN PRINT AS ONE CHUNK, BUT LONG ONES
     ARE SPLIT IN TWO
4050 A$="right"+LEFT$(Z$(I),128):B$="right"+RIGHT$(Z$(I),L-128):REM
     ADD CURSOR RIGHT TO PRINT CORRECTLY TO TAPE
4060 PRINT #1,A$: PRINT #1,I: GOSUB 5000: PRINT #1,B$: GOTO 4100
4080 A$="right"+Z$(I):PRINT #1,A$
4100 NEXT I
4120 PRINT #1,999:PRINT #1, 999: REM FILE END MARKER TWICE FOR SAFETY
4200 CLOSE 1: PRINT "FILE WRITTEN": GOTO 80
5000 POKE 59411,53: T=TI: REM TURN ON TAPE MOTOR
5020 IF (TI-T)< 6 GOTO 5020: REM FOR .1 SECONDS
5040 POKE 59411,61:RETURN:REM TURN OFF MOTOR
```



Rate-Setting and Billing for Small Utilities

BY STEPHEN P. SMITH



This article presents the development and implementation of a package which includes billing and record keeping routines for a small utility, in this case a town water system. We'll see how the computer assists in establishing water rates, how it maintains records, prepares the bills and provides statistical data to monitor the distribution of the water. The modest hardware requirement can be imitated using almost any microcomputer. The BASIC software, with the possible exception of audio tape utilities, can be used directly on any system with an 8K interpreter.

The water system in question is operated by the town of Parksley, Virginia. Parksley has a population of about a thousand people and a prosperous business section servicing the surrounding rural area. Water is drawn from several deep wells and pumped to a tower for storage. Gravity feeds water from the tower to business and residential customers. Although a large portion of the system was built using Federal money, it is now maintained from fees and town property taxes.

Historically, these fees have been a flat $9 per household per quarter. The result was that Parksley used almost twice as much water per capita as did similar towns with metered systems. Households that conserved water subsidized the water wasters, and there was no incentive to repair leaks. Also, the system's total capacity was nearly reached, while the need for new connections was steadily increasing. As an alternative to expensive captial improvements, the town elected to install water meters, charge customers by the gallon, and hope that consumption would be forced down.

This conversion presents two problems. The first is to determine a rate schedule that will achieve an adequate income. The current solution is to set the minimum rate at the existing $9 per quarter, but with the addition of a computer it will be possible to further reward conservation-minded households without jeopardizing the solvency of the system. The second problem is the actual preparation of the bills. Meters must always be read by hand, but manual preparation of bills has proved to be time consuming, costly and error prone. The task is ideal for automation, but people tend to fear that impersonal and inflexible billing, characteristic of large utilities, will result. This is an area where the intimacy and personal control possible with a microcomputer will prove to be a convincing factor.

In addition to solving these problems, automation will provide a bonus. With all the meter readings in its memory, the computer can do a fairly thorough statistical analysis. Consequently, the town can see the average use for residential and business customers, along with the deviations from month to month. The town will have a good idea what range of use can be expected at various times of the year. A statistically significant increase at any one meter might indicate a leak or some other problem with water use. Finally, customers who question the billing procedures can be shown exactly how their consumption compares with that of others in the town.

Illustration by David Bastille

How Much Per Gallon?

Let's begin by looking at the technique used to establish rates. This is basically a one-time program, but it might be used again should use patterns of water use change in the near future. The town has a good idea of what revenues it needs to operate the system, and knows the total amount of water pumped from the wells. From government surveys, we can also determine how much water a typical household should use and how the price of water would affect that use. What we need is a model, a set of equations that will accept this data and find the optimum rate scale.

The model developed here is the Monte Carlo type. That is, it generates its own random population based on

given statistics, then tries various rate schedules and compares their effects. We don't really know how many people live in each house in Parksley or how much water they use. We do know, however, what percentage of households (on a regional average) have 1, 2, 3, . . . , people and we know the normal range of water use for a household of any given size. We also know, from data published by the U.S. Department of Commerce, how a change in a water bill will effect consumption. We can't predict the behavior of individuals, but we can be fairly certain about the average effect over a population the size of Parksley's.

Our procedure, therefore, is to generate a random population whose water use pattern matches that expected for a town of 1000 persons. Next, we bill them according to a specific schedule and see what the effect will be on their consumption. Bills are then generated for the new consumption and we determine if the revenues and total water use meet our goals. If not, the schedule is adjusted; a new consumption is calculated; and the entire process is repeated. After several iterations, the procedure produces a table of rates consistent with good water management and fiscal need.

The first step is to generate the population. The computer does this using the BASIC random number function. The first set of numbers assigns the size of each household. About 3% of the possible random numbers will assign one person to a house; 50% will assign four persons. The percentages are provided to the program in data statements. The same statements contain the normal range of use per person in households of each given size. A second series of random numbers will be used to select a specific rate of consumption within this range. Different random number generators will pick different populations, and none may be exactly like that found in Parksley. Provided that our statistics are good, however, each will produce almost identical results in our billing model. These results demonstrate the power of the Monte Carlo technique.

The next step is to prepare a bill for each water user we have created. For Parksley, a five-level schedule has been selected with the hope that it will produce enough flexibility to control waste without overly confusing the customers. (The local electric power co-op also uses a five-tier schedule.) The cutoff for each level is a function of use and is determined by the program. The approximately 100 customers whose use is lower than one standard deviation below the mean (see the accompanying box on normal distributions) pay only the minimum connection fee. Consumption within each standard deviation above that is billed at a successively higher rate, until the 10% of the customers whose water use is more than two standard deviations above the mean are paying the highest rate. Within each level, the price per 1000 gallons is determined by an equation of the form C1 + N * C2. N is the number of the price level (from 0 to 4). As we will see, this formula allows the rate schedule to be corrected for either low revenue or high consumption.

Many businesses, including small utilities, often must vary rates by use. Computers can simplify calculations needed to insure equitable rates.

Finally, we can use the bills we have just generated to determine what the new water use patterns will be. Information provided in data statements allows our program to calculate the consumption each household should reach within a few billing cycles. The billing algorithm is run once again and the resulting revenue is compared with that needed to run the system. If too little money is brought in, constant C1 in the rate is increased. If too much money (say a 10% surplus) is generated, C1 is reduced, lowering prices across the board. If the new total consumption is too high, constant C2 is raised, increasing prices for heavy users. The program now returns to the original consumption data and calculates bills using the new schedule. Once again the effect of the bills consumption is found and the rate schedule is revised accordingly. The procedure is repeated over and over, until both fiscal and conservation criteria are met.

Does This Apply to My Business

This type of trade-off analysis should be familiar to most business people. If your prices are high, you make a good profit on each sale, but expect few sales. If your prices are low, you get more sales, but less profit on each. The optimum is the point at which total profit is the greatest. Although the town only needs to break even, the problem is similar. A rate schedule too highly weighted for conservation might reduce revenue below that needed to run the system. A schedule weighted too lightly might allow use to exceed the system's total capacity. Finally, if the overall rate structure is too high, the citizens may revolt and vote out the town officers at the next election.

This Monte Carlo model can be adapted for any small business enterprise. Market surveys will supply data on how customers will react to various price levels. Often the Federal government or industry groups will have already published this work. A good manager will know how his costs will vary with production or purchase levels, sales volume, warehousing, taxes on inventory and changes in the need for personnel and equipment. Given this data, a computer model will be able to select the optimum price for a product, predict demand and even show how altering price structure could change your share of the market (perhaps lowering current profits, but providing an even larger return in the future).

The Billing Program

The rate schedule generated with the Monte Carlo model becomes one of the inputs to the billing routine. The other inputs are the meter readings and a historical data file. This file includes last month's readings and certain statistical data for each meter. The new readings are entered at a terminal by the town secretary. All that is needed from the software is a simple prompting routine and some checks for obvious errors. The historical data must come from mass storage; and it's here that microcomputers most often fail to perform adequately for business applications.

Without good system utilities, it is very difficult to build disk or tape files that will cover any situation. Because we are limiting our computer system to this single task, however, we can handle the chore with an inexpensive audio tape recorder and a little BASIC software.

Having established that tape will be used, we must now determine what in-

formation is needed on the file. Certainly the last meter reading must be included. To help track changes in consumption, the last month's water use and the average use at this meter will appear. Additional statistical parameters such as standard deviation may also prove helpful in predicting use patterns. Each record on the tape will be identified by a unique meter number, and also by the name and address of the person receiving the bill. Reading and updating this tape involves a trade off. Reading the entire tape before doing any processing requires a lot of memory. This method is often the only alternative with the primative hardware available for microprocesssor systems. Adding a little special purpose hardware, however, we can provide our program with control of the tape recorder motor and allow it to read the file one record at a time. Unfortunately, if each reocrd is read and processed individually, the operator will spend a great deal of time waiting for the relatively slow audio tape to enter its data.

The procedure which has proven most advantageous in this application involves software motor control and two readings of the tape which extract needed data without the operator being present. The tape is first read before any new data are entered. The system stores the meter number, the previous reading and the last monthly consumption. The other data are ignored. About 6000 words of memory are required for this array. As the operator enters new readings, the billing routine calculates the number of gallons used and compares it with the figure for the previous month. Unusual changes are flagged so that the operator may check for input errors or make notes to examine the meter location for problems.

When all the new readings have been entered, the mean and standard deviation of water use for the entire town are calculated. These figures are used to measure progress toward the consumption goal and to assist in future rate calculations. They may also be used to tell individual users how their consumption compares to that of their neighbors. (Think of the effect of receiving a bill telling you you're using more water than 97% of your neighbors!)

The second reading of the tape takes place automatically. A single record is read and all the data are retained in memory. The computer now prints the bill for this customer, addressing it as shown on the data tape. The operator can store a list of customer name and address changes before the process starts. The list is keyed to meter numbers and corrects the tape as the bill writing proceeds. As each bill is printed, a new tape is written. This tape becomes the input file for the next month's billing procedure.

Because a relatively slow audio tape is used for storage and a very slow KSR-35 teletype is used for printing, this billing procedure is a lengthy process. The system has proven exceptionally reliable, however and can be run unattended. The operator keys in the new readings on the day they are collected, sets up the tapes and then leaves. The computer runs overnight, preparing bills and making a new tape.

Hardware

I'm sure that most data processing managers would be horrified at the prospect of using such equipment to do their work. From the standpoint of cost effectiveness, however, I doubt that they can match it.

The hardware consists of an Ohio Scientific Challenger II with 12K words of user memory and 8K Microsoft BASIC in ROM, a used KSR-35 Teletype terminal, two Kansas City standard audio cassette recorders and some relays for motor control. The entire package costs less than $1500. Additions like disk drives and a high speed printer would be nice, but they are clearly not necessary and may contribute to the user's fear of the computer.

Overcoming this fear is one of the most important functions of the personal computer. Many potential business applications users have no experience with data processing, and out of ignorance, distrust the computer. The Parksley water billing system shows what can be done to encourage new users. The same hardware could just as easily perform inventory or general ledger operations. Alternately, any other hardware configuration (for example, your personal system) could do this job. The key to success is the development of software geared directly to the user's application. With this power available, a business person faced with paying tens of thousands of dollars for a general purpose minicomputer will be very interested in a custom tailored micro costing much, much less.

The Normal Distribution

Operation of the Monte Carlo model relies not on knowing about a specific group of people, but on the fact that we can know how people behave in general. They are described to the computer in terms of statistics. For example, 50% of all households have four people, and the average daily water use is 120 gallons per person.

In building a sample population, however, we must know more than just the average. We must know if all the elements of our population should be close to the average, or if they should be scattered widely. Perhaps a large number just below the average should be balanced by a few who are far above it. This kind of information is supplied by a statistical distribution.

The distribution most commonly used is called simply the normal distribution. It characterizes almost any type of measurement, from test scores to the value of resistors to the water use of a community. Theoretically, the normal distribution appears as the bell-shaped curve shown in Figure 1. The majority of the population will be grouped at the center, near the average or mean. As you leave the center in either direction you find fewer and fewer samples.

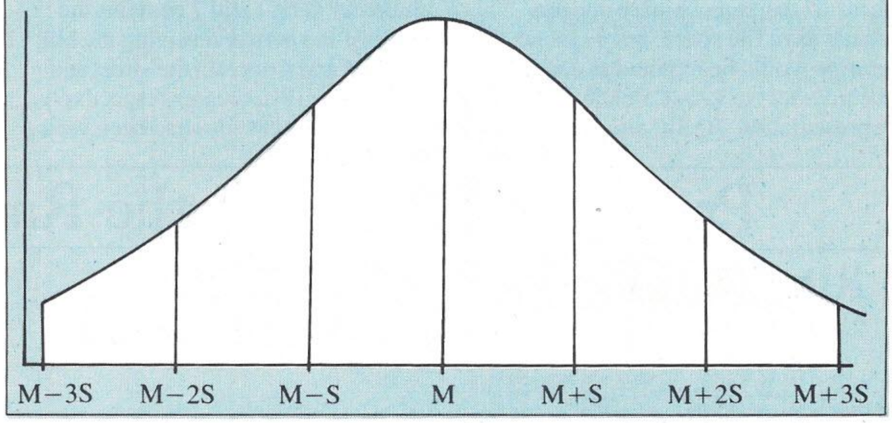
Just how closely the samples group around the mean is called the central tendency. In a normal distribution, the measure of central tendency is called the standard deviation: 68% of a normal population falls within one standard deviation of the mean, 34% above and 34% below. Another 28% lie between one and two standard deviations, and most of the remaining 14% lie between two and three. In each case, half are above and half are below the mean.

Suppose we know, for example, that the mean test grade in a class of 50 pupils was 85% and that the standard deviation was 5%. We can be fairly certain that 17 of the grades fell between 85 and 90, and that 17 more were between 80 and 85. It is unlikely, although not impossible, that more than

one pupil scored better than 95. We can also predict how many grades fell between any two limits. All these calculations are taken from the mean, standard deviation and the mathematical definition of the normal distribution.

The mean and standard deviation are calculated from real data such as census surveys or (in other applications) samples of a product. Often only these statistical parameters, and not the original data, are available. Using tables or equations that describe the distribution, however, we can create a new model population that will behave in approximately the same manner as the original. Thus, we can determine how a town will behave without ever meeting its inhabitants.

Figure 1 Bell-Shaped Curve



M is the mean or average. S is the standard deviation. The height of the curve represents the probability of finding element of the population with the given value.

Controlling Audio Tape From BASIC

The relatively low speed of the audio tape used for mass storage in the water billing system makes manual control of the tape drives impractical. If an operator were required to turn the recorders on and off for each record, any savings which might be realized through automation would quickly be lost. It is necessary, therefore, to provide computer control. Fortunately most quality cassette recorders have a remote on/off capability. Our software only needs access to the switch.

The Ohio Scientific 430 I/O board which provides the audio tape port in this system has several unused output lines which can perform the switch function. High order address lines are decoded to activate the board. The three low order address lines are demultiplexed with the Read/Write line to produce 16 lines which strobe (go briefly from the zero to one level) when the correct memory location is accessed. Thus, PEEK and POKE commands in BASIC will activate one of the strobe lines.

Two flip-flops are added to latch these strobes. Accessing memory location D10N or D20N causes the corresponding latch to be set. Accessing DOFF causes the corresponding latch to be set. Accessing DOFF resets both latches. The latched outputs drive small relays which actually control the current to the tape recorders.

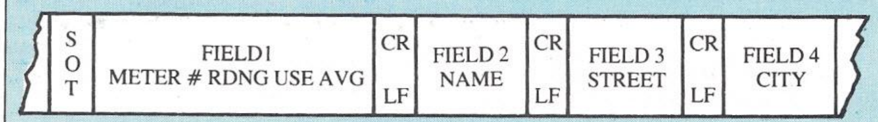
File Structure

Producing a computer system for a specific application such as water billing requires more than choosing a computer and writing some BASIC software. Careful consideration must be given to the format in which data are stored. You, as the system analyst, must decide which data to save, the number of digits to present, the units to use, and whether to store numeric values as strings of ASCII characters, as binary numbers or in BASIC floating point representation. In many cases you must make a trade off between an efficient file structure and the amount of money you're willing to invest in mass storage devices for your system.

Data for the water billing system are stored in a single tape file. The file contains a series of records. Each record begins with an SOT (start of text) character and contains four field of data. (See Figure 2.) The fields are separated by a carriage return and line feed. Because data are accessed sequentially (one character at a time) and because special characters are used as delimiters, the data fields may be of variable length. Only necessary characters are included, producing the shortest records. This method is of considerable importance when slow mass storage such as audio tape is used.

The first of the four data fields contains four numeric values separated by commas. These values are all integers and will fit nicely in the six significant figure accuracy of the Microsoft BASIC. The first value is the meter number and serves to identify the record. The second is the last meter reading (gallons*10); the third is the previous usage (gallons); and the last is the

Figure 2



Sequential file holds meter reading and name and address of owner.

weighted average of all previous uses. This field is read as a string of characters and then converted to numeric form. Using direct binary integers would save file space, but a special routine would be required to convert the numbers to BASIC floating point representation. BASIC already has a function VAL(X), to convert from string notation.

The other three fields remain as character data. Field 2 contains the name of the person receiving the bill. Fields 3 and 4 contain the street and city/state/ZIP addresses respectively. If random access files had been used (e.g., on floppy disk), the name and address data might have been stored separately from the meter data, allowing each file to be updated independently. The use of sequential access, however, dictated the combined format as most effective. The file structure had to be designed around hardware selection.

Program 1 – Setting the Rate Schedule

Run

```
ESTIMATION OF RATE SCHEDULE TO ENCOURAGE WATER CONVERSATION
TARGET COMSUMPTION = 909091  GALLONS PER MONTH
REVENJE SHOULD FALL BETWEEN $ 300  AND $ 327.273

A POPULATION OF 100  ELEMENTS HAS BEEN CREATED
 TOTAL USAGE = 1.08754E+06 GALLONS PER MONTH
MEAN USAGE = 10875.4
STANDARD DEVIATION = 4405.22
A = .227003    K = .227003
PREDICTED CONSUMPTION = 1.04265E+06       REVENUES = $ 277.26
A = .267945    K = .256082
PREDICTED CONSUMPTION = 1.02285E+06       REVENUES = $ 314.57
A = .267945    K = .423157
PREDICTED CONSUMPTION = 999198            REVENUES = $ 317.67
A = .267945    K = 1.05959
*PREDICTED CONSUMPTION = 910416           REVENUES = $ 282.73
A = .310153    K = 1.06909
*PREDICTED CONSUMPTION = 891854           REVENUES = $ 308.24
RECOMENDED RATE SCHEDULE
LESS THAN 6470.16 GALLONS - $ 2.00674
 6470 TO 10870  ADD $ .310153  PER 1000 GALLONS
 10870 TO 15280  ADD $ 1.37042  PER 1000 GALLONS
 15280 TO 19680  ADD $ 2.4307  PER 1000 GALLONS
MORE THAN 24090 ADD $ 3.49097 PER 1000 GALLONS
```

This routine finds a rate scale which should cause total water consumption to be reduced to the target level while insuring that revenues will be adequate to operate the water works. A Monte Carlo model creates a user population based on the population characteristic of the town and U.S. Department of Commerce statistics on residential water consumption. Bills are calculated using a proposed rate scale. The probable effect of increased bills is then computed and a new total consumption and total revenue are calculated. If either consumption or revenue do not meet the set criteria, the rate schedule is adjusted and the bill/effect/bill/test procedure is repeated.

A similar technique can be applied to marketing or production problems, but the user should be careful. This interactive technique may not converge on a solution when two variables are adjusted independently as in this program. It is important to have some insight into the problem and design a correction (i.e., change in rate schedule) compatible with its effect (i.e., change in consumption). The equation used here cannot be considered general, but are designed to insure convergence to a reasonable solution.

Listing

```
1 PRINT"ESTIMATION OF RATE SCHEDULE TO ENCOURAGE WATER CONVERSATION"
2 REM WRITTEN BY STEPHEN P. SMITH
3 REM            STATE COLLEGE, PA
4 REM MICROSOFT BASIC
5 P=100
10 DIM U(P),U1(P),V(P),B(P),B1(P)
11 DIM D(6),UU(6),UL(6),N(6)
12 MINREV=825*P/275
13 MAXREV=900*P/275
14 TARGET=2500*P/275
15 PRINT"TARGET COMSUMPTION ="TARGET*1000" GALLONS PER MONTH"
16 PRINT"REVENUE SHOULD FALL BETWEEN $"MINREV" AND $"MAXREV
17 PRINT
19 GOSUB 200: REM CREARE A MODEL POPULATION
20 S=S2:M=M2
22 A=1/S:K=A
24 PRINT"A ="A,"K ="K
25 GOSUB 460: REM BILL AT CURRENT USAGE
30 GOSUB 700: REM FIND EFFECT OF HIGHER BILLS
35 GOSUB 460: REM PREDICT NEW BILLS
36 PRINT"PREDICTED CONSUMPTION ="UT*1000,"REVENUES = $"INT(BT*100)/100
40 D=-1
42 FOR I=1 TO P:U(I)=U1(I):NEXT
45 K2=K:U2=UT
48 REM ADJUST K FACTOR TO MEET TARGET
50 K=K+ABS((K-KL)/(UT-UL))*(UT-TARGET)
52 KL=K2:UL=U2
54 IF UT>TARGET THEN D=0
55 IF UT<0.95*TARGET THEN D=0
58 REM IF REVENUE TOO LOW, RAISE A FACTOR
60 IF BT<MINREV THEN A=A*MAXREV/BT:D=0
68 REM IF REVENUE TOO HIGH, LOWER A FACTOR
70 IF BT>MAXREV THEN A=A*MINREV/BT:D=0
80 IF NOT D GOTO 24
84 REM
85 REM ************************************
90 PRINT"RECOMENDED RATE SCHEDULE"
100 PRINT"LESS THAN"(M-S)*1000"GALLONS - $"A*(M-S)
110 FOR I=0 TO 2
120 PRINT INT((M+(I-1)*S)*100)*10"TO"INT((M+I*S)*100)*10" ADD $"A+I*K;
121 PRINT " PER 1000 GALLONS"
130 NEXT
140 PRINT"MORE THAN"INT((M+3*S)*100)*10"ADD $"A+3*K"PER 1000 GALLONS"
190 STOP
194 REM
195 REM ************************************
200 REM MODEL SUBROUTINE
210 REM DATA TABLES HOLD PERCENT OF POPULATION WITH N PERSONS/FAMILY
212 REM FOLLOWED BY UPPER AND LOWER DAILY USE PER PERSON
220 DATA 3,30,100,13,40,120,33,40,140,83,60,150,97,50,130,100,40,120
230 FOR I=1 TO 6:READ D(I),UL(I),UU(I):NEXT
240 FOR I=1 TO P
250 X=RND(1)*100
260 FOR J=1 TO 6
270 IF X>D(J) THEN NEXT J
275 REM SET FAMILY SIZE
290 V(I)=J
300 N(J)=N(J)+1
305 REM SET USE BY HOUSEHOLD
310 U(I)=(UL(J)+(UU(J)-UL(J))*RND(2))*30/1000*J
315 REM RUNNING TOTAL
320 T=T+U(I)
330 S2=S2+U(I)*U(I)
340 NEXT I
345 REM MEAN (AVERAGE)
350 M2=T/P
355 REM STANDARD DEVIATION
360 S2=SQR((S2-P*M2*M2)/(P-1))
370 PRINT"A POPULATION OF"P" ELEMENTS HAS BEEN CREATED"
400 PRINT" TOTAL USAGE ="T*1000"GALLONS PER MONTH"
410 PRINT"MEAN USAGE ="M2*1000
415 PRINT"STANDARD DEVIATION ="S2*1000
420 RETURN
430 REM
440 REM ************************************
460 REM BILLING SUBROUTINE
462 REM CALCULATE BILL FOR EACH METER BASED ON PROPOSED SCHEDULE
470 BT=0:UT=0
480 FOR I=1 TO P
490 B(I)=A*(M-S)
495 UT=UT+U(I)
500 S1=INT((U(I)-M)/S)+2
520 IF S1<=0 GOTO 610
525 S2=-1
530 IF S1>4 THEN S1=4
550 U1=U(I)-M+S
560 S1=S1-1
565 S2=S2+1
570 IF S1=0 THEN B(I)=B(I)+U1*(A+S2*K):GOTO 610
580 B(I)=B(I)+S*(A+S2*K)
590 U1=U1-S
600 GOTO 560
610 BT=BT+B(I)
612 NEXT I
620 RETURN
630 REM
640 REM ************************************
700 REM EFFECT SUBROUTINE
702 REM EFFECT OF HIGHER BILL HAS THREE FACTORS
710 M2=M:S2=S
720 FOR I=1 TO P
760 U1(I)=U(I)
765 B1(I)=B(I)
768 REM 1ST FACTOR IS INCREASE IN BILL
769 REM CURRENT BILL IS 3 DOLLARS PER MONTH
770 DB=B(I)-3
775 IF DB<=0 GOTO 830: REM LOWER BILLS WON'T RAISE CONSUMPTION
778 REM 2ND FACTOR IS POTENTIAL FOR DECREASE
780 DU=U(I)/V(I)-UL(V(I))*30/1000
790 REM 3RD FACTOR IS RANDOM AND SCALING
800 DF=(1+RND(3))*.3
810 DU=DU*DB*DF
820 U(I)=U(I)-DU
824 TEST=UL(V(I))*30/1000*V(I)
825 IF U(I)<TEST THEN PRINT"*";:U(I)=TEST
830 NEXT I
840 RETURN
```

Program 2 – Billing Routine

Run

```
DATE? "SEPTEMBER 30,1978
AUTO ENTRY? YES
METER # 23456
READING? 346789
METER # 23473
READING? 235654
*****************
STATISTICAL DATA FOR TOWN,  SEPTEMBER 30,1978
TOTAL  2198  MEAN  1099  STD DEV  0
*****************
CHANGE OF USER OR ADDRESS
METER #? 0
IS TAPE 2 SET TO RECORD HEADER? YES
DATE BILLS DUE? "OCTOBER 31, 1978
INPUT TAPE ON DRIVE ONE
OUTPUT TAPE ON DRIVE TWO
READY? YES


TOWN WATER SERVICE
PARKSLEY, VA 23421
SEPTEMBER 30,1978


                    STEPHEN P. SMITH
                    106 EAST CLEARVIEW
                    STATE COLLEGE, PA 16801


FOR METER 23456 READING 345678 TO 346789
PLEASE REMIT $2.00 ON OR BEFORE OCTOBER 31, 1978


THIS BILL WAS PREPARED BY COMPUTER
BUT ERRORS MAY BE REPORTED TO A PERSON AT 665-5090.
NEW LOWER RATES FOR CAREFULL USERS
```

```
TOWN WATER SERVICE
PARKSLEY, VA 23421
SEPTEMBER 30,1978


                    STEPHEN P. SMITH
                    106 EAST CLEARVIEW
                    STATE COLLEGE, PA 16801


FOR METER 23473 READING 234567 TO 235654
PLEASE REMIT $2.00 ON OR BEFORE OCTOBER 31, 1978


THIS BILL WAS PREPARED BY COMPUTER
BUT ERRORS MAY BE REPORTED TO A PERSON AT 665-5090.
NEW LOWER RATES FOR CAREFULL USERS
```

```
BILLING COMPLETE FOR 2 METERS
NEW METERS MAY BE ADDED TO SYSTEM
IS  OUTPUT TAPE READY ON DRIVE TWO? NO
```

This program computes and prints the bills for over 200 residential and commercial water meters. All data are contained in a single sequential file on audio cassettes. The file is updated by making a new copy.

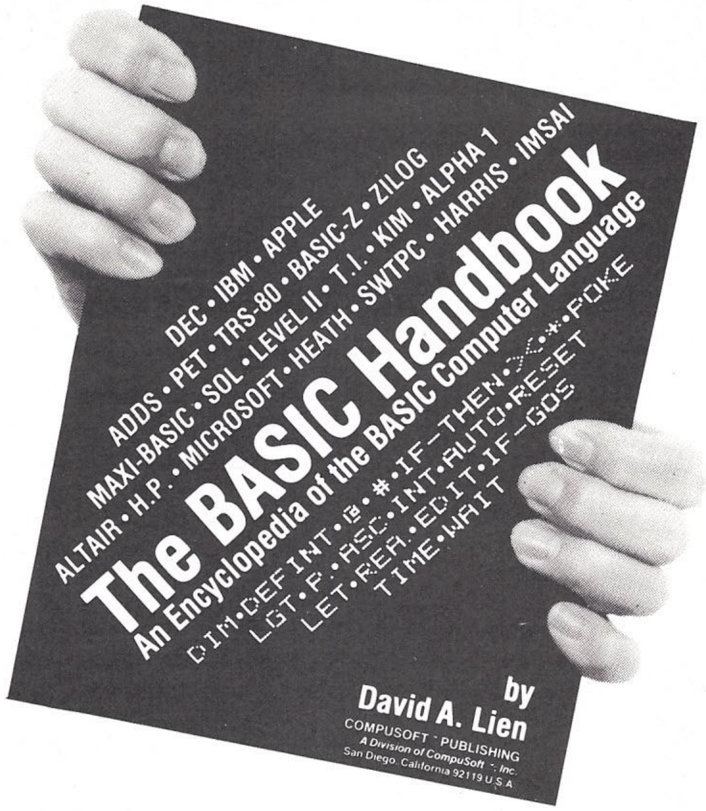
The program begins by reading the entire file. Next, the operator is prompted to enter the new meter readings. A check insures that the data are at least reasonable. When all new readings are entered, the operator may store address changes for use when the updated file is written. The updates occur as the file is read again, this time record by record. With each read, a fee is calculated, a bill is printed, and a new record is written. Finally, the operator may add new meters to the updated file.

The program is a good example of top down design. Although it relies on Ohio Scientific's implementation of audio tape to operate, modification to suit other hardware should be easy. Note also how the extended variable names permitted by Microsoft's interpreter make the code more readable.

Listing

```
10 REM MODEL UTILITY BILLING PROGRAM
12 REM MICROSOFT BASIC AND OSI 430 TAPE I/O
14 REM
16 PO=275: REM NUMBER OF METERS IN SYSTEM
20 DIM M(PO),R(PO),R1(PO),U(PO),U1(PO),A(PO)
30 DIM M2(10),R2(10),N$(10),S$(10),C$(10)
40 CR$=CHR$(13): REM CARRIAGE RETURN
50 SOT$=CHR$(2): REM START OF TEXT
60 FF$=CHR$(12): REM FORM FEED
65 LF$=CHR$(10): REM LINE FEED
70 D1ON=64258: REM STROBE FOR DRIVE ONE ON
80 D2ON=64260: REM STROBE FOR DRIVE TWO ON
90 DOFF=64263: REM STROBE FOR BOTH DRIVES OFF
100 FRACTION=.25: REM POSSIBLE CHANGE AT ONE METER IN ONE MONTH
110 TAPE=64259:REM TAPE PORT (FB03 STATUS REG,FB05 I/O PORT)
140 INPUT"DATE";DATE$
150 GOSUB 310: REM READ TAPE FOR LAST MONTH'S DATA
160 INPUT"AUTO ENTRY";A$
170 IF A$="YES" THEN GOSUB 1210:GOTO 185: REM METER # GENERATED AUTOMA
180 GOSUB 1270: REM OPERATOR ENTERS METER NUMBERS
185 GOSUB 1610: REM STAISTICS
190 GOSUB 1710: REM CHANGE OF ADDRESS
200 GOSUB 2210: REM WRITE HEADER
210 GOSUB 2410: REM DO BILLS
220 PRINT"BILLING COMPLETE FOR"PO"METERS"
230 GOSUB 2810: REM ADD NEW METERS
240 END
290 REM**********************
300 REM
310 REM READ TAPE
320 INPUT"NUMBER OF RECORDS ON TAPE";PO
330 PRINT"SET DRIVE 1 TO PLAYBACK"
340 INPUT "READY";A$
350 FOR I=1 TO PO
360 GOSUB 510: REM GET A RECORD
370 M(I)=METER
380 R(I)=REEDING
390 U(I)=USAGE
400 A(I)=AVERAGE
410 NEXT
420 RETURN
490 REM**********************
500 REM
510 REM READ A RECORD
520 X=PEEK(D1ON): REM START DRIVE ONE
530 GOSUB 910: REM GET CHAR$
540 IF CHAR$<>SOT$ GOTO 530
550 REM FOUND START OF RECORD
560 GOSUB 810: REM GET LINE
570 METER$=LINE$
580 GOSUB 810: REM GET LINE
590 NAME$=LINES$
600 GOSUB 810: REM GET LINE
610 STREET$=LINE$
620 GOSUB 810: REM GET LINE
630 CITY$=LINE$
640 X=PEEK(DOFF): REM TAPE OFF
650 REM DECODE METER$
660 LINE$=METER$
670 GOSUB 1010:REM GET NUMBER
675 METER=NUMBER
680 GOSUB 1010: REM GET NUMBER
690 REEDING=NUMBER
700 GOSUB 1010: REM GE T NUMBER
710 USAGE=NUMBER
720 GOSUB 1010: REM GETNUMBER
730 AVERAGE=NUMBER
740 RETURN
790 REM**********************
800 REM
810 REM GETLINE
815 LINE$=""
820 GOSUB 910: REM GET CHAR$
830 IF CHAR$=CR$ THEN RETURN
840 IF ASC(CHAR$)>31 THEN LINE$=LINE$+CHAR$
850 GOTO 820
890 REM**********************
900 REM
910 REM GET CHARACTER IN CHAR$
920 WAIT TAPE,128,128: REM TEST STATUS REGISTER
930 X=PEEK(TAPE+2)
940 CHAR$=CHR$(X): REM DATA STORED AS ASCII CHARACTERS
950 RETURN
990 REM**********************
1000 REM
1010 REM GET NUMBER FROM LEFT SIDE OF STRING LINE$
1020 L=LEN(LINE$)
1030 NUM$=""
1040 FOR I=1 TO L
1050 CHAR$=MID$(LINE$,I,1)
1060 IF CHAR$=" " GOTO 1090
1070 NUM$=NUM$+CHAR$
1080 NEXT
1090 NUMBER=VAL(NUM$)
1100 LINE$=RIGHT$(LINE$,L-I)
1110 RETURN
1119 REM*********************
1200 REM
1210 REM  ENTER METER NUMBERS AUTOMATICALLY
```

CompuSoft™ Publishing Proudly Announces . . . . . .



$14.95

(360 pages - Soft Cover)

From the author of the highly acclaimed TRS-80 Learner's Manual comes the book you've been asking for! THE BASIC HANDBOOK is THE definitive reference and "idea" book which explains in detail the BASIC language as used in over 50 micros, minis and mainframes.

Not a dictionary, not a text, it is a virtual ENCYCLOPEDIA of the BASIC language. Explaining all you need to know about over 250 BASIC statements, functions, operators and commands, it is the "missing link" needed to convert programs from one computer to RUN on another.

YES, I need the BASIC Handbook!

COMPUSOFT PUBLISHING 8643 PC Navajo Road, San Diego, CA 92119

Please send ____________ copies of The BASIC Handbook. My check for $14.95 each + $1.35 P&H is enclosed. (CA Addresses add 6% tax)

I understand my order will be shipped promptly and there is a 30 Day money-back guarantee.

Name ______________________

Address ______________________

City ______________________ Zip __________

My computer is a ______________________

CIRCLE 30

Listing 2 continued

```
1220 AUTO=-1
1230 FOR I=1 TO PO
1240 METER=M(I)
1250 PRINT"METER #"METER
1260 GOSUB 1350: REM INPUT READING ETC.
1262 NEXT
1264 RETURN
1265 REM***********************
1268 REM
1270 REM OPERATOR ENTERS METER NUMBERS
1290 INPUT"METER #";METER
1300 IF METER=0 THEN RETURN
1310 FOR I=1 TO PO
1320 IF METER=M(I) THEN GOSUB 1350: GOTO 1290
1330 NEXT
1340 PRINT"INVALID METER NUMBER. TYPE 0 TO RETURN":GOTO 1290
1349 REM***********************
1350 REM
1352 REM INPUT READING ETC
1354 INPUT"READING";REEDING
1360 USAGE=REEDING-R(I)
1370 DELTA=ABS(USAGE-U(I))
1380 AVERAGE=A(I)
1385 IF DELTA<FRACTION*USAGE OR DELTA<FRACTION*AVERAGE GOTO 1500
1390 PRINT"CURRENT USAGE"USAGE"IS OUT OF PROBABLE RANGE"
1400 PRINT"LAST USAGE="U(I);"LAST READING"R(I);"AVERAGE USAGE="A(I)
1430 INPUT"WAS READING IN ERROR";A$
1440 IF A$="YES" GOTO 1350: REM GET CORRECTED READING
1500 REM UPDATE RECORD WITH CORRECT READING
1510 U1(I)=U(I):U(I)=USAGE
1520 R1(I)=R(I):R(I)=REEDING
1530 A(I)=(USAGE+AVERAGE*3)/4: REM RUNNING WEIGHTED AVERAGE
1560 RETURN
1590 REM**********************
1600 REM
1610 REM STATISTICS
1620 S=0:T=0
1630 FOR I=1 TO PO
1640 T=T+U(I): REM TOTAL USAGE FOR TOWN
1650 S=S+U(I)*U(I)
1660 NEXT
1670 M=T/PO: REM MEAN USAGE AT ALL METERS
1680 S=SQR((S2-PO*M*M)/(PO-1)): REM STANDARD DEVIATION
1682 PRINT"*****************
1684 PRINT"STATISTICAL DATA FOR TOWN,",DATE$
1685 PRINT"TOTAL "T" MEAN "M" STD DEV "S
1686 PRINT"*****************
1690 RETURN
1699 REM 0*************************
1700 REM
1710 PRINT"CHANGE OF USER OR ADDRESS"
1720 FOR I=1 TO 10
1725 GOSUB 1920: REM INITIALIZE METER
1730 IF METER=0 GOTO 1820
1740 M2(I)=METER
1760 R2(I)=REEDING
1770 N$(I)=NAME$
1780 S$(I)=STREET$
1790 C$(I)=CITY$
1800 NEXT
1810 PRINT"STORAGE EXHAUSTED *** PROGRAM ERROR #12
1820 RETURN
1890 REM**************************
1900 REM
1910 REM INITIALIZE METER
1920 INPUT"METER #";METER
1930 IF METER=0 THEN RETURN
1940 INPUT"READING";REEDING
1950 USAGE=0
1960 AVERAGE=12000
1970 INPUT"NAME";NAME$
1980 INPUT"STREET";STREET$
1990 INPUT"CITY";CITY$
1995 GOTO 1920
1999 REM******************************
2000 REM
2010 REM WRITE A RECORD
2040 FOR I=1 TO 1000:NEXT: REM  RECORD A BLANK LEADER
2050 CHAR$=SOT$:GOSUB  3410: REM PUT CHAR$
2060 LINE$=STR$(METER)+STR$(REEDING)+STR$(USAGE)+STR$(AVERAGE)
2065 GOSUB 3310: REM PUT LINE
2070 LINE$=NAME$: GOSUB 3310: REM PUT LINE
2080 LINE$=STREET$: GOSUB 3310: REM PUT LINE
2090 LINE$=CITY$: GOSUB 3310: REM PUT LINE
2100 LINE$="": GOSUB 3310: REM PUT LINE
2110 FOR I=1 TO 1000:NEXT: REM TRAILING BLANK TAPE
2120 X=PEEK(DOFF): REM ALL DRIVES OFF
2130 RETURN
2190 REM*************************
2200 REM
2210 REM WRITE HEADER
2220 INPUT"IS TAPE 2 SET TO RECORD HEADER";A$
2230 IF A$="NO" GOTO 2220
2240 X=PEEK(D2ON): REM DRIVE TWO ON
2250 FOR I=1 TO 1000:NEXT: REM LEADER
2260 LINE$="RECORDS UPDATED ON "+DATE+" FOR "+STR$(PO)+" EXISTING "
2265 GOSUB 3310: REM PUT LINE
2270 LINE$="TOTAL USAGE LAST PERIOD WAS "+STR$(T)+" GALLONS"
2275 GOSUB 3310: REM PUT LINE
2280 LINE$="MEAN USAGE LAST PERIOD "+STR$(M)
2285 GOSUB 3310: REM PUT LINE
2290 LINE$="STANDARD DEVIATION "+STR$(S)
2295 GOSUB 3310: REM PUT LINE
2300 LINE$="TOTAL BILLING $"+TB
2305 GOSUB 3310: REM PUT LINE
2310 LINE$="": GOSUB 3310: REM PUT LINE
2390 REM*************************
2400 REM
2410 REM DO BILLS
2420 PRINT"INPUT TAPE ON DRIVE ONE"
2430 PRINT"OUTPUT TAPE ON DRIVE TWO"
2440 INPUT"READY";A$
2450 FOR II=1 TO PO
2460 GOSUB 510: REM READ A RECORD
2470 GOSUB 3210: REM ENTER CHANGES
2472 GOSUB 3010: REM FIND FEE
2474 GOSUB 3510: REM DOLLARS AND CENTS
2475 TB=TB+AMOUNT
2480 REEDING=R(II)
2490 USAGE=U(II)
2500 AVERAGE=A(II)
2505 REM
2506 REM PRINT BILL
2510 PRINT FF$: REM FORM FEED
2520 PRINT"TOWN WATER SERVICE"
2530 PRINT"PARKSLEY, VA 23421"
2540 PRINT DATE$
2550 PRINT:PRINT
2560 PRINT TAB(20);NAME$
2570 PRINT TAB(20);STREET$
2580 PRINT TAB(20);CITY$
2590 PRINT:PRINT
2600 PRINT "FOR METER"METER"READING"R1(II) "TO"R(II)
2610 PRINT"PLEASE REMIT $"AMOUNT$" ON OR BEFORE "DUE$
2620 PRINT:PRINT
2630 GOSUB 2910: REM COMMENTS
2640 GOSUB 2010: REM WRITE A RECORD ON TAPE
2650 NEXT II
2670 RETURN
2690 REM**********************
2800 REM
2810 PRINT"NEW METERS MAY BE ADDED TO SYSTEM"
2820 INPUT"IS  OUTPUT TAPE READY ON DRIVE TWO";A$
2830 IF A$="NO" THEN RETURN
2840 GOSUB 1920: REM INITIALIZE METER
2850 IF METER=0 THEN RETURN
2860 GOSUB 2010: REM WRITE A RECORD
2870 GOTO 2830
2890 REM************************
2900 REM
2910 REM COMMENTS
2920 REM THIS ROUTINE CAN ADD A MESSAGE ON EVERY BILL
2925 REM
2930 PRINT"THIS BILL WAS PREPARED BY COMPUTER"
2940 PRINT"BUT ERRORS MAY BE REPORTED TO A PERSON AT 665-5090."
2945 REM
2950 REM IT CAN ALSO ADD MESSAGES TO SPECIFIC BILLS
2955 REM
2960 IF AMOUNT<3 THEN PRINT"NEW LOWER RATES FOR CAREFULL USERS"
2970 RETURN
2990 REM**********************
3000 REM
3010 REM FIND AMOUNT
3020 IF KNOWN GOTO 3070
3030 DIM LEVEL(5),RATE(5)
3040 DATA 8000,2,12000,3,16000,4,20000,5,24000,6
3050 FOR I1=1 TO 5:READ LEVEL(I1),RATE(I1):NEXT I1
3060 KNOWN=1
3070 AMOUNT=RATE(1): REM MINIMUM BILL
3080 IF USAGE<LEVEL(1) THEN RETURN
3090 FOR I2=2 TO 5
3100 IF USAGE<LEVEL(I2) THEN AMOUNT=AMOUNT+(USAGE-LEVEL(I2-1))*RATE(I2)
 3105 IF USAGE<LEVEL(I2) THEN RETURN
3110 AMOUNT=AMOUNT+(LEVEL(I2)-LEVEL(I2-1))*RATE(I2)
3120 NEXT I2
3130 AMOUNT=AMOUNT+(USAGE-LEVEL(5))*RATE(5)
3140 RETURN
3190 REM*********************
3200 REM
3210 REM ENTER CHANGES
3220 FOR I1=1 TO 10
3230 IF METER<>M2(I1) GOTO 3280
3240 NAME$=N$(I1)
3250 STREET$=S$(I1)
3260 CITY$=C$(I1)
3270 GOTO 3290
3280 NEXT I1
3290 RETURN
3299 REM**********************
3300 REM
3310 REM PUT LINE
3320 L=LEN(LINE$)
3330 FOR I2=1 TO L
3340 CHAR$=MID$(LINE$,I2,1)
3350 GOSUB 3410: REM PUT CHAR$
3360 NEXT I2
3370 CHAR$=CR$: GOSUB 3410: REM PUT CHAR$
3380 CHAR$=LF$: GOSUB 3410: REM PUT CHAR$
3390 RETURN
3399 REM**********************
3400 REM
3410 REM PUT CHAR$
3420 WAIT TAPE,128,128
3430 POKE TAPE+2,ASC(CHAR$)
3440 RETURN
3450 REM**********************
3500 REM
3510 REM FORMAT AMOUMT AS DOLLARS AND CENTS IN AMOUNT$
3520 X1=AMOUNT
3525 IF X1<1 THEN X0$="0"
3530 X1=INT(X1*100+0.5)/100
3540 X0$="":X$=""
3550 IF X1=0 GOTO 3630
3630 X1$=STR$(X1)
3635 X2=LEN(X1$)
3640 X2=X2-1
3645 X1$=MID$(X1$,2,X2)
3650 FOR I=1 TO X2
3655 X3=I
3660 X2$=MID$(X1$,I,1)
3665 IF X2$="." THEN 3685
3670 NEXT I
3675 X$=".00"
3685 IF X3=(X2-1) THEN X$="0"
3690 AMOUNT$=X0$+X1$+X$
3700 RETURN
```

WE'RE ALTOS COMPUTER SYSTEMS. Our SUN-SERIES ACS8000 business/scientific computer creates a new standard in quality and reliability in high technology computers.

HIGH TECHNOLOGY The ACS8000 is a single board, Z80®* disk-based computer. It utilizes the ultra-reliable Shugart family of 8 inch, IBM compatible, disk drives. A choice of drives is available: single or double density, single or double sided. Select the disk capacity you need, when you need it: ½M, 1M, 2M, or 4M bytes. The ACS8000 features the ultimate in high technology hardware: a fast 4 MHz Z80 CPU, 64 kilobytes of 16K dynamic RAM, 1 kilobyte of 2708 EPROM, an AMD 9511 floating point processor, a Western Digital floppy disk controller, a Z80 direct memory access, Z80 Parallel and Serial I/O (two serial RS232 ports, 1 parallel port), and a Z80 CTC Programmable Counter/Timer (real time clock). In essence, the best in integrated circuit technology.

BUILT-IN RELIABILITY The ACS8000 is a true single board computer. This makes it inherently reliable and maintainable. The board and the two Shugart drives are easily accessible and can be removed in less than five minutes. All electronics are socketed for quick replacement. Altos provides complete diagnostic utility software for drives and memory.

QUALITY SOFTWARE Unlimited versatility. The ACS 8000 supports the widely accepted CP/M®** disk operating system and FOUR high level languages: BASIC, COBOL, PASCAL and FORTRAN IV. All available NOW.

PRICE ACS 8000-1, single density, single-sided [½ Mb] $3,840
ACS 8000-2, double density, single-sided [1 Mb] $4,500
ACS 8000-3, single density, double-sided [1 Mb] $4,800
ACS 8000-4, double density, double-sided [2 Mb] $5,300

Brackets show disk capacity per standard two drive system. All models come standard with 32 Kb RAM and two 8″ disk drives as shown above. Expansion to 64 Kb is $363 per 16 Kb. FPP, DMA, software optional. Dealer/OEM discounts available. Delivery: 30 days ARO, all models.

*Z80 is a trademark of Zilog, Inc.
**CP/M is a trademark of Digital Research, Inc.

Circle 7 on inquiry card.

ALTOS
COMPUTER SYSTEMS 2338A WALSH AVENUE • SANTA CLARA • CA 95050 • (408) 244-5766

Tired of Typing GOTO? Try T-Bug

BY WILLIAM L. COLSHER

Bored with Level I BASIC on your TRS-80, but haven't got the $99 that Level II costs? Why not give T-BUG a try. You've probably been telling yourself that you really should learn machine language anyway, right? So why not give up next week's beer money and spring the $14.95 it costs for a copy?

T-BUG, a Radio Shack program tape, allows you to enter machine language programs directly into your TRS-80. And machine language allows you to create some really spectacular games. It also lets you write very efficient programs (or user routines, if you've got Level II) to do things much faster than even the best-written BASIC program could. You have full control of the computer's input/output capabilities, so you can operate some of those really neat peripherals that are available — or even design your own.

The only catch to all this is that you have to know or learn machine language. This is not nearly as difficult as it sounds, and there are a number of good beginner's books to help. If you can write programs in BASIC, you shouldn't have too much trouble learning assembler. Just take it slow and work carefully.

Radio Shack assumes that you've got at least a fundamental knowledge of assembly language. The only materials they supply with T-BUG are a ten-page "User Instruction Manual" on T-BUG itself and a copy of the *Mostek Z-80 Pocket Reference*. The documentation is at least a little better than what we've become accustomed to in this hobby; but it seems to be rather hastily done, with a number of grammatical errors left in to annoy the reader.

T-BUG itself is 1K (1028) bytes long including a stack area. That leaves the user with about 3K in the minimal system. You can do a *lot* with 3K. That 3K is all program space, too. The display area is completely separate.

T-BUG is no harder to get going than a BASIC program; it loads with the CLOAD command. Once it's loaded, the prompt (#) appears and you're ready to enter commands.

T-BUG's nine commands allow you to enter, examine, debug, load and save machine language programs. Each command is entered into the system in the form of a one-character abbreviation (see Table 1). Commands can only be entered when the prompt is the first character on the line.

Let's take a look at each command and what it does. Clearly, the most important command is the one that allows the user to examine and change locations in memory. Since this command affects *M*emory locations, the abbreviation to be entered is "M". Following the "M" you must enter the hexadecimal address of the first memory location you wish to examine or change. T-BUG then displays the contents of that location on the screen. At this point you have three options. You can enter a two-digit hex number. This number is then stored at the addressed location and the next address and the contents of that address are displayed. Your second option is to hit the ENTER key which leaves the current location the same and displays the next location. The final option is to hit the "X" key. This returns you to the prompt and lets you enter any other command.

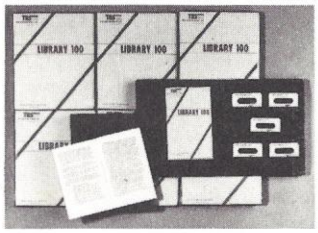
Once you've entered a program and checked it over, you obviously want to execute it. To do so, use the "J" command (short for *J*ump). After entering the "J", type in the four-digit hexadecimal address of the first instruction of your program. Control is then trans-

Table 1

Command	Function
B xxxx	Insert a break point at the hex address xxxx
F	Restore the original instruction after a break point
G	Begin execution at the location of a "restored" break point. (Used after an "F")
J xxxx	Jump to the location specified by the hex address xxxx
L	Load a program or data tape created by the "P" command
M xxxx	Display the contents of the memory location specified by the hex address xxxx. The contents of the location can be changed at will.
P xxxx yyyy	Save the program or data from location xxxx to location yyyy on tape. Both addresses are in hex.
R	Display the contents of the CPU registers in hex
X	Terminate the execution of a B, J, M or P instruction before typing the last digit.

COMPLETE
BACKUP.

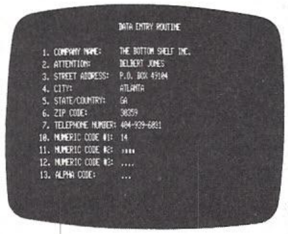
Library 100—$49.50



The Library 100 was designed to fulfill your general TRS-80 Level II programming needs. Using advanced Level II techniques and rigid quality control, The Bottom Shelf has been able to make available 100 programs on five guaranteed-to-load cassettes, which load over a wide range due to advanced recording techniques and methods designed by engineers for The Bottom Shelf. The programs include applications in five areas: Business and Finance, Education, Graphics, Home and Games.

As an added bonus, you get Tiny PILOT, the first new high-level language for the TRS-80. It's perfect for teachers, parents, students and sales trainees.

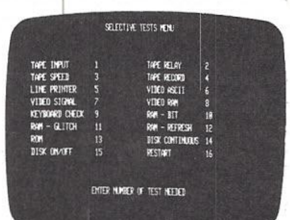
Mail List—$125.00



For your 32K, two disk drive and printer TRS-80, an endless mailing list!

MAIL LIST keeps 500 names and addresses per disk, together with three numeric codes and one alphabetic code for each name. 500 records per disk, and any number of disks that you wish. MAIL LIST keeps all entries in zip code and then last name order, and keeps track of which disk is holding which items. Automatic prompts instruct the operator as to which disk should be inserted. Selectable printouts for labels or other lists for hard copy storage. Full EDIT, DELETE, and other commands for your operators convenience. See SYSTEMS EXTENSIONS or your dealer for full details. **The first TRS-80 batch-processing program.**

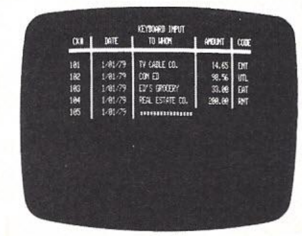
System Doctor—$28.50



Now you can check your TRS-80 Level II computer and all it's peripherals with this program. SYSTEM DOCTOR allows you to selectively test any function or peripheral or continuously test your system. Run this program once a week or when trouble occurs. Be your own computer diagnostician.

Checkbook II—$18.50

By Alan Meyers



The technical expertise of TBS has been synthisized into a program that upgrades checkbook analysis to an interesting, informative and fun portion of your regimen. This program outshines the other programs of its type to such a degree that it is a new category of checkbook analysis for the consumer.

Featuring:
- Five column keyboard input handling amounts up to $100,000.
- Complete editing mode allowing changes in any or all columns plus a deletion module.
- Checkbook balancing with balance brought forward always in memory and recorded on tape.
- Bank reconcile with listing of all outstanding checks. Automatically deletes all cleared checks upon completion.
- Search and Total from all checks in memory or an entire month by all fields but amount.
- 16K version holds 75 transactions while the 32K holds 400!

The Basic Toolkit—$19.80

By F. Barry Mulligan

A Programmer's Solution. The Basic Toolkit is a dream come true! This machine language program will do the following:
- Search a basic program, and print out to screen or printer, an alphabetized listing of the variables used in the program and the line numbers in which they are found.
- Search and print to screen or printer, a listing of all GOTO's and GOSUB's and the line numbers in which they appear.
- Restore basic programs that have been accidentally lost (after typing NEW or WHEN YOU GO TO DOS).
- Will check for bad memory in 15 seconds.
- WILL MERGE PROGRAMS ON A CASSETTE BASED SYSTEM!
- Will search memory for all occurences of a specific byte and list the locations where it appeared.

See your local TBS dealer, over 100 nationwide. Master Charge or Visa accepted. All orders plus $2.00 P & H. Add 4% tax, if Georgia. Above software for TRS-80 16K L II.

TBS INC.

The Bottom Shelf, Inc.
P.O. Box 49104
Atlanta, Georgia 30359
Phone: 404-939-6031

CIRCLE 33

ferred to that location by means of an unconditional jump.

Most of us have at least a couple of bugs left in our programs when we enter them for the first time, so Radio Shack has thoughtfully provided a "break" command to help with debugging. Abbreviated "B", this command is also followed by the hex address of the instruction you wish to stop execution at. This address must be the address of a one-byte instruction to the address of the first byte of a multi-byte instruction. T-BUG then saves the original contents of that location and replaces it with a three-byte instruction that returns control to T-BUG. This feature enables you to examine the registers and data locations and figure out what the program's problem is.

After you've figured out what's wrong, you can restore the original instruction (the one replaced by the break) simply by pressing the "F" key on the keyboard. You can then press the "G" key and the program will start up with the instruction that had been replaced by the break. Obviously, you can only put one break at a time in the program since it saves the instruction that it replaces and you could quickly run out of space without some kind of limit.

I mentioned above that you might want to look at the contents of the CPU registers. This is easily done with the "R" command. T-BUG will display the registers in a rather difficult to read format. I've become used to the very impressive display that the Digital Group monitor provides and T-BUG's is rather disappointing for that reason. There's no labeling of the registers on the screen — just a bunch of hex numbers. You have to refer to the user's manual until you get used to the format.

I also have two other complaints about the display. First, the Interrupt Vector register is not displayed; and, second, the Flag register (F) is displayed in hexadecimal. You have to break it down into binary to determine what a given flag is set to.

Once you've gotten a program to work, you'll probably want to save it on tape. Type a "P" followed by the beginning and ending addresses (in hex) of the program to be saved. Assuming that you have your tape recorder hooked up right (just as if you were saving a BASIC program), T-BUG will write your program out to tape in a format compatible with the LOAD command. Radio Shack provides a description of this format in the manual.

The load command, abbreviated "L", works just like CLOAD in BASIC. Just type the command; if the cassette recorder is ready, T-BUG will load the program or data from the tape. Just as in BASIC, an asterisk blinks in the upper left hand corner of the screen to assure you that loading is going all right. In case of error, an "E" is displayed to the right of the "L". If this happens, you must try to reload the tape from the beginning.

If you're at all like me, you will occasionally make a typing mistake. If you do, the "X"command terminates whatever command is currently running — generally before the last digit is entered. Of course, since the "L" and "F" commands start up as soon as you type the character, "X"won't work with them.

As a 1K monitor, T-BUG is fairly typical. There's only so much you can fit into 1K. I already mentioned that I think the register display could be improved. That's really the only weak point of any importance.

The rest of my complaints concern the documentation. I don't like to dig through machine code one byte at a time to figure out where useful routines are hidden. Radio Shack has documented the keyboard input routine as well as the TV output routine and the cassette I/O routines in ROM (Read Only Memory). It would have been nice if they'd told us a little more. One example of a useful routine that isn't mentioned is the code that handles the TRS-80 BASIC language arithmetic.

The biggest omission, though, is the graphic characters. There are a bunch of graphic characters that you can access from machine language; but the documentation doesn't tell you how.

No doubt someone will come up with a much better monitor soon; but for the money T-BUG is the best you can get right now. It also has the advantage of being available in your local Radio Shack, so you don't have to wait on the mail. If you've got any interest in using all the capabilities of that grey box of yours, by all means pick up a copy of T-BUG and dive right in. □

"HOT" NEW BOOKS FOR MICRO-COMPUTER PROS!

The latest in micro-computer books that people waited in line to buy!
Now available by mail for the first time.

How to Buy & Use Minicomputers & Microcomputers
By W. Barden, Jr.
Explores the basics; covers hardware, software, peripheral devices, programming languages & techniques; use for fun or profit! 240 pgs. *#21351 . . . $9.95*

Microcomputer Primer
By M. Waite & M. Pardee
Explains it all: central processing unit, memory, input/output interfaces, programs. Discusses well-known models & shows how to program your own. 224 pgs. *#21404 . . . $7.95*

Basic Primer
By M. Waite & M. Pardee
Covers BASIC fundamentals, program control, organization, functions & variations. With a game program & data on numbering systems & ASCII character codes. 192 pgs. *#21586 . . . $8.95*

How to Program Microcomputers
By W. Barden, Jr.
Examines assembly-language programming of 8080, MC6800 & MCS6502 models. Explains concepts, operation & architecture. With precanned operations. 256 pgs. *#21459 . . . $8.95*

Fundamentals of Digital Computers (2nd Ed.)
By D. Spencer
Explains computers, programming, hardware, BASIC, FORTRAN & COBOL languages, advanced concepts & computer center operation. 320 pgs. *#21534 . . . $9.95.*

The 8080A Bugbook®: Microcomputer Interfacing and Programming
By Rony, Larsen & Titus
Emphasizes the computer as a controller. Covers principals & 4 basic tasks of computer interfacing. BUGBOOK® is a registered trademark of E&L Instruments, Inc., Derby, Conn. 06418. 416 pgs. *#21447 . . . $10.50*

DBUG: An 8080 Interpretive Debugger
By C. Titus & J. Titus
How to use DBUG (a software debugging package) to develop & test programs. With documented hexadecimal & octal listings. 112 pgs. *#21536 . . . $4.95*

Introductory Experiments in Digital Electronics and 8080A Microcomputer Programming and Interfacing
By Rony, Larsen & Titus
Covers basic gates thru microcomputer interrupts, with hardware & software examples. Experiments re: breadboarding, circuits, etc. over 416 pgs. each. Book 1: *#21550 . . . $12.95.* Book 2: *#21551 . . . $10.95.* Both: *#21552 . . . $20.95*

NCR Data Communications Concepts
By NCR Corp.
Explores concepts, techniques & limitations of data communications: telephone-based & others. 208 pgs. *#21548 . . . $6.95*

NCR Data Processing Concepts Course
By NCR Corp.
Intro to data processing, computers & digital logic. Covers input/output devices, memories, control. 256 pgs. *#21547 . . . $7.95*

Understanding CMOS Integrated Circuits (2nd Ed.)
By R. Melen & H. Garland
Begins with basic digital ICs; covers semiconductor physics, CMOS fabrication technology & design, & advanced CMOS applications. 144 pgs. *#21598 . . . $5.95*

Transistor-Transistor Logic (2nd Ed.)
By G. Flynn
Discusses digital logic & different types of logic circuits used in the TTL functional class. With block diagrams, schematics, logic truth tables. 288 pgs. *#21572 . . . $6.95*

SAVE 10% when you order 3 or more!
Examine any book(s) at NO RISK for 15 Days!

315

Clip Out–Mail Today!

YES—Please send me the book(s) indicated below. If I'm not completely satisfied, I may return any or all within 15 days of receipt for full credit or refund. Add sales tax where applicable.

☐ 21351	☐ 21534	☐ 21552
☐ 21404	☐ 21447	☐ 21548
☐ 21586	☐ 21536	☐ 21547
☐ 21459	☐ 21550	☐ 21598
	☐ 21551	☐ 21572

☐ I have checked 3 or more titles and deducted my 10% savings.

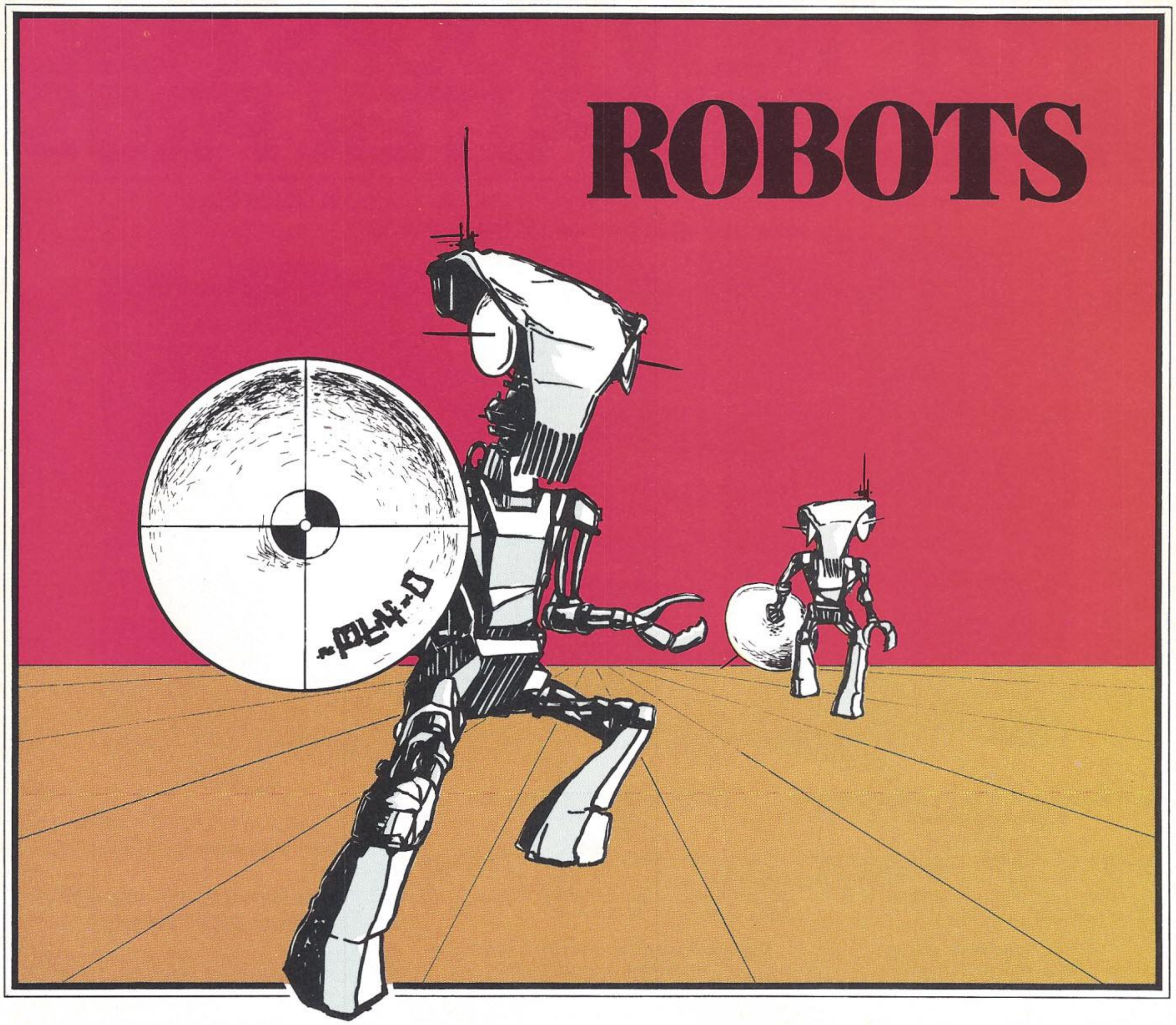
Name ________ (Please Print)
Address ________
City ________
State ________ Zip Code ________ Total: $ ________

☐ Check ☐ Money Order ☐ Master Charge ☐ Visa/BankAmericard
Exp. Date ________
Account No.: ________
Interbank No.: ________ (Master Charge only)

Signature: ________
Minimum credit card purchase: $10.00 MF51

MAIL TO:
Howard W. Sams & Co., Inc.
4300 W. 62nd St.
Indianapolis, IN 46206

Prices subject to change 6 months after issue date.



ROBOTS

BY WILLIAM LAPPEN

You are in a room with killer robots. Through some cosmic oversight, the robots have only heat sensors (no eyes). They are programmed to seek you out and destroy you (by occupying the same square you are on — like chess). Their creator did not anticipate that the room in which the chase takes place also has electronic force beams that don't give off any heat. If a robot moves into one of these force beams, it will be destroyed. The same goes for you. Further, if two robots collide with each other, both will be destroyed. The object of the game is to position yourself so that all of the evil robots are destroyed by the beams or crash into each other.

Movements allowed to you and the robots are similar except that you may stay still but the robots must move towards you. The moves are the same as the king's in a chess game (horizontal, vertical or diagonal). You move one space and then *all* of the robots move one space towards you. Some will hit the walls (force beams) and others will collide. But they do converge on you pretty quickly.

You can probably modify this game program to play in real time. Radio Shack Level I BASIC does not have a command that allows the computer to input a character from the keyboard as soon as it is typed. So this version is played with each side (you and the robots) taking turns.

This program has been designed to run as quickly as possible. As in most time/memory trade-offs, the program is very inefficient in memory use. For example, an array almost 900 elements long just keeps track of what has been printed on the screen; and only even numbered elements are ever used in the array as described below.

Basically, the computer must keep track of the locations of the force beams (twice as many as robots) and the robots, along with the location of the opponent. From a memory point of view, the most efficient method would be to keep one array of information and break it into "stationary" (walls) and "mobile" (robots and opponent) sections. Then, when each move is computed, the whole array up to the element being moved would be searched for something else already on that array. The program currently runs in a Radio Shack TRS-80 with 16K RAM. It requires about 8K but can be made to fit into 4K by searching through the

Illustration by David Bastille

location array instead of the screen square. Searching past the element moved is unnecessary because you are only interested in what is on that square at the end of the turn. For example, if one robot is trailing another, it would be wrong to eliminate both of them just because the back robot moves forward before the front robot can get out of its way. This would leave collisions partly up to the luck of which robot happens to move first.

With this program, a game with 20 robots and 40 walls would have to search the first 40 elements of the array 20 times plus each element, I, of the mobile section 20 - I times. This search is exceedingly slow.

A refinement could be made by keeping the two sections sorted in ascending order. The walls section would only have to be sorted once. The time taken to locate a wall on the square or another robot that has moved onto the square would be less; but you must either sort the mobile section of the array at the end of each turn, or rebuild the array in sorted order, sorting after each robot's move (insertion sort). While this method does not seem to take long, it ruins the "pseudo-animation" effect that can be achieved with huge arrays. Nonetheless, this method is acceptable for 4K machines.

With more memory, representing the whole screen with an array is faster. This way, the computer would only check one element of the array to see if there has been a crash (either with another robot or into a wall). I used this approach. The array is "A" from 64 to 960. (Radio Shack Level I only allows one array, but it may be very large.) Instead of searching the long "screen" array to find out where the robots are when ready to move them, I have stored that information in the first 63 elements of the array. (A game will only rarely use over 35 robots.) Because walls do not move, only the locations of robots are stored in this array. As a further speed factor, this "location" array only contains the location of active robots. Once a robot is destroyed, an active robot is moved into its place and the end pointer is decremented.

Turning to the actual code, lines 40 to165 print out the instructions. Then the "screen" array is reset. Due to visual considerations, I chose to use every other position on the horizontal line (only even numbered positions). Otherwise, horizontal motion looks shorter than vertical motion.

PRINT AT S		S =
490		10040
800		10040
1400		10040
4400	previous	3900
4530		3900
5915	previous	5900
5970	previous	5900
6140		5900
6180		5900
6205		5900
6208		5900
6250		5900
6254		5900
6300		5900
30020		5900
30050		5900
30060		5900
30068		5900

Table 1

With the Radio Shack print statement, you can PRINT AT S where S (or any variable or constant) is a position on the screen. Instead of using X and Y coordinates, Tandy chose to number each position consecutively from 0 to 1023. If your computer allows you to use X and Y coordinates, substitute those into the print statement. Table 1 contains the PRINT AT S command and its equivalent X and Y coordinates. "Previous" means that the number was computed on the last move and is now stored (you are moving *from* a square). If you print using X and Y coordinates, I suggest that you save the locations the same way using a two dimensional array. This method saves conversion time. Conversion is Y=INT (S/64+.01): X=X - Y*64.

Back to the program. At line 200, the user is asked to enter a difficulty level (number of robots) if this is the first pass through the program. Otherwise, D will be greater than 0, control will skip around this segment, and the program will adjust the next game based on the performance during the last. If the opponent won the game, the next will have two more robots and four more walls. If the opponent lost, the program will take into account the number of remaining robots when computing a new difficulty level.

The boundaries of the room are then drawn. The SET command uses X and Y coordinates to turn on a block of light. There are six such blocks per alphanumeric position on the screen (1024*6 for the whole screen).

Next, the program randomly assigns locations to the opponent, then the robots and then the walls. The assignment subroutine, lines 10000 to 10060, will not let a subsequent pick replace a former one. So, if the game puts a robot next to you, there is no chance that a wall will later replace it! By filling the room in this order, the games are a little more challenging than by using a reverse order fill.

The opponent's location is stored in A(0) while the robots' locations are stored in A(1) to A(D). At line 3000, the program is ready to move the opponent and awaits an instruction. (The "+" signs in lines 3500 to 3800 stand for "OR".) Once the move is broken into its X and Y components, converted to a number from 64 to 960 and verified, the program moves and updates starting at line 4460.

Because I keep more than one array, I have to signal when I have moved one of the robots. Therefore, before moving any robots, I change their numbers in the screen array from 1 to 0.5. This signals that the robot is yet to be moved. As long as the computer has to loop through all of the elements of the location array, it might as well perform the "repacking" function at this point. When an inactive robot is discovered, the last element is moved to that location and the location array is shortened. This way, by line 5050, the first C elements of the location array have the locations of all active robots.

The move for each robot is now computed, while the opponent's move is broken into X and Y components, converted to a screen number and then verified. Then it is moved and the arrays updated. The timing loops (starting at lines 5950, 6160, 6206, 6252, 6255, 30030, 30053 and 30062) provide some animation. For example, at 6160, the robot has been printed where a wall was. The program pauses just long enough to let the user see the robot "move into the wall" and then reprints the wall.

ROBOTS is not only entertaining, but also illustrates memory/speed trade-off dramatically. Instead of searching many elements of an array, you can represent the whole screen in one large array. □

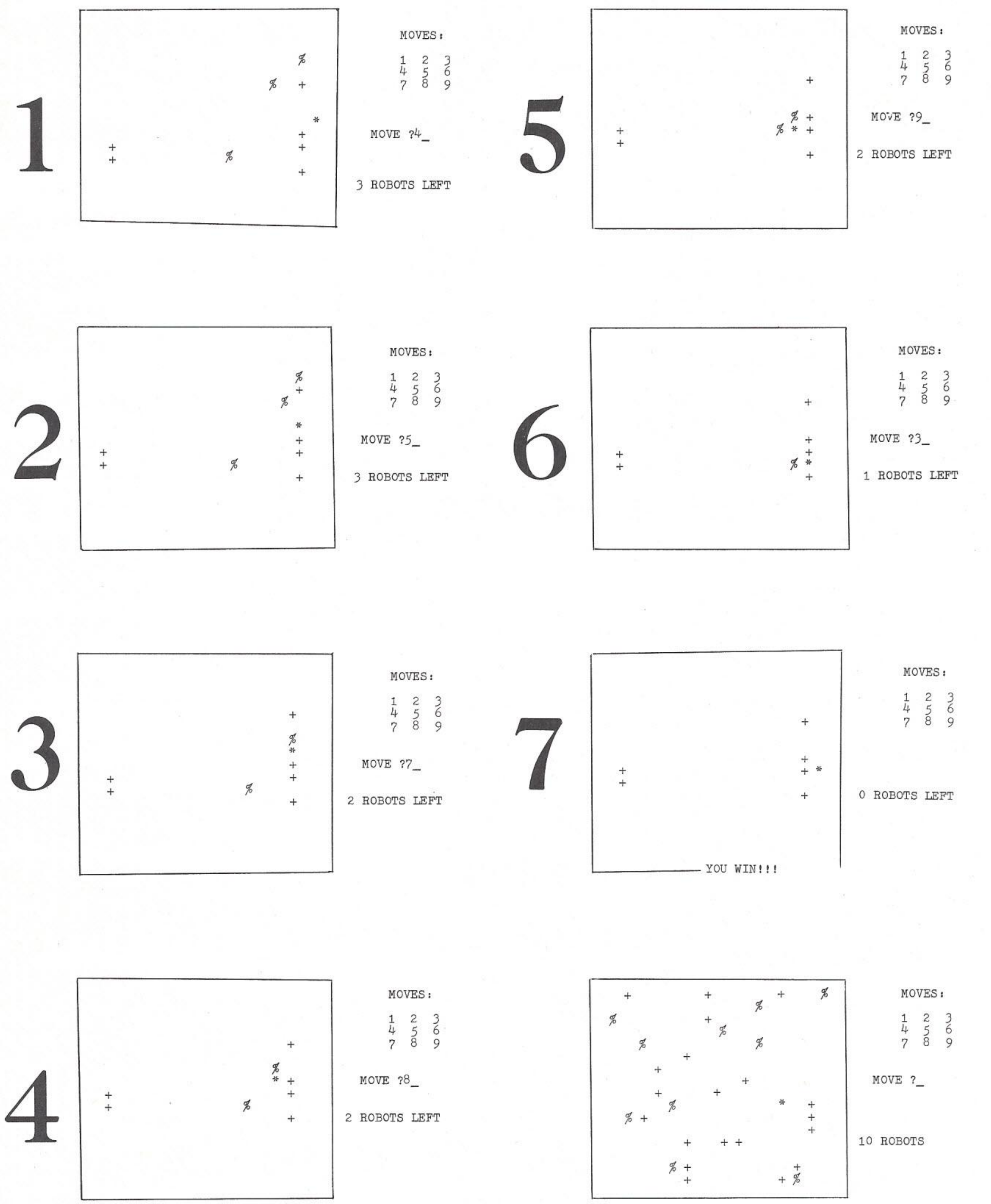
Program Listing

```
10  REM ROBOTS 7/26/78
20  D=0
30  CLS
40  PRINT "YOU ARE IN A ROOM WITH KILLER ROBOTS AND ELECTRONIC"
50  PRINT "WALLS.  THE ROBOTS ARE EQUIPPED WITH HEAT SENSORS"
60  PRINT "AND ARE PROGRAMMED TO DESTROY YOU.  IF THEY (OR YOU)"
70  PRINT "HIT THE WALLS, INSTANT DEATH RESULTS.  FURTHER, IF"
80  PRINT "TWO ROBOTS COLLIDE, THEY WILL BOTH BE DESTROYED."
90  PRINT "YOU ARE THE '*', THE ROBOTS ARE THE '%' AND THE"
100  PRINT "WALLS ARE THE '+'.  YOU MAY MOVE IN ANY DIRECTION"
110  PRINT " AND THE MOVES ARE NUMBERED:"
120  PRINT TAB (20); "1  2  3"
130  PRINT TAB (20); "4  5  6"
140  PRINT TAB (20); "7  8  9"
150  PRINT "FOR EXAMPLE, 1 WOULD MOVE YOU NORTHWEST AND 5"
160  PRINT "WOULD LEAVE YOU WHERE YOU ARE.  GOOD LUCK!!!"
163  PRINT
165  PRINT
166  REM RESET A(64-960) WHICH REPRESENTS THE SCREEN
170  FOR I=64 TO 960 STEP 2
180  A(I)=0
190  NEXT I
200  IF D>0 D=D+2-INT(F/2):  GOTO 405
300  INPUT "DIFFICULTY LEVEL ";D
400  IF (D<1) + (D>40) GOTO 300
405  CLS
407  C=D
408  REM DRAW THE ROOM
410  FOR I=4 TC 92
420  SET (I,2)
425 NEXT I
430  FOR I=2 TO 45
433  SET (92,I)
436  SET (93,I)
439  NEXT I
440  FOR I=92 TO 4 STEP -1
450  SET (I,45)
455  NEXT I
460  FOR I=45 TO 3 STEP -1
463  SET (4,I)
466  SET (5,I)
469  NEXT I
470  GOSUB 10000
475  REM S HAS THE LOCATION OF THE *
480  A(S)=2
485  A(0)=S
490  REM FILL ROOM WITH D ROBOTS AND RECORD LOCATION IN A(1-D)
495  PRINT AT S; "*";
500  FOR I=1 TO D
600 GOSUB 10000
800  PRINT AT S; "%";
900  A(S)=1
950  A(I)=S
1000 NEXT I
1050 REM FILL ROOM WITH 2*D WALLS
1100 FOR I=D+1 to D*3
1200 GOSUB 10000
1400 PRINT AT S; "+";
1500 A(S)=-1
1600 NEXT I
1650 REM A(0) HAS LOCATION OF *
1660 REM A(64-960) REPRESENTS THE SCREEN.  1 IS A ROBOT, 2 IS
1670 REM THE *, -1 IS A WALL, AND 0 IS A BLANK.
1680 REM A(1-D) HAS LOCATION OF ROBOTS (%).
1700 PRINT AT 184; "MOVES:"
1800 PRINT AT 312; "1  2  3"
1900 PRINT AT 376; "4  5  6"
2000 PRINT AT 440; "7  8  9"
2500 V=D
2510 REM V IS THE NUMBER OF ACTIVE ROBOTS
2530 PRINT AT 815; V; "ROBOTS LEFT";
2550 IF V=0 GOTO 25000
2600 F=V
3000 PINT AT 628; "MOVE ";
3050 INPUT M
3100 IF (M<1) + (M>9) GOTO 3000
3200 PRINT AT 628
3300 REM READY TO MOVE *
3350 REM BREAK MOVE (M) INTO ITS X AND Y COMPONENTS
3400 X=0
3420 Y=0
3450 IF M=5 GOTO 4600
3500 IF (M=1) + (M=4) + (M=7) X=-1
3600 IF (M=3) + (M=6) + (M=9) X=1
3700 IF (M=1) + (M=2) + (M=3) Y=-1
3800 IF (M=7) + (M=8) + (M=9) Y=1
3900 S=A(0)+Y*64+X*2
3950 REM S IS THE NEW LOCATION
4000 A=INT(S/64+.001)*64
4100 IF (S-A<3) + (S-A>45) GOTO 3000
4200 A=INT(S/64+.001)
4300 IF (A<1) + (A>14) GOTO 3000
4350 REM HAVE VALID MOVE
4400 PRINT AT A(0); "";
4450 IF A(S)<>0 X=-3: GOTO 30000
4460 REM UPDATE LOCATION AND SCREEN ARRAYS
4470 A(A(0))=0
4500 A(0)=S
4530 PRINT AT A(0); "*";
4560 A(S)=2
4600 REM READY TO MOVE ROBOTS
4700 V=0
4800 I=0
4850 I=I+1
4860 REM SET ALL ROBOTS TO .5 IN SCREEN ARRAY TO SIGNAL THAT
```

Robots Listing continued

```
4870 REM THE ROBOT HASN'T YET BEEN MOVED
4900 IF (A(I)<>0) * (A(A(I))>=.5) A(A(I))=.5: GOTO 5000
4905 REM ROBOT I IS NOW INACTIVE
4910 A(I)=A(C)
4920 C=C-1
4930 I=I-1
5000 IF I<C GOTO 4850
5050 REM C IS THE NUMBER OF ACTIVE ROBOTS
5070 REM BREAK VECTOR INTO X AND Y COMPONENTS
5100 FOR I=1 TO C
5250 A=A(0)-INT(A(0)/64+.001)*64:B=A(I)-INT(A(I)/64+.001)*64
5300 A=A-B
5400 B=A(0)-A(I)
5405 IF B<0 B=B-4
5410 IF B>0 B=B+4
5415 IF B<>0 G=B/ABS(B)
5420 B=INT(ABS(B)/64+.001)*G
5450 IF A=0 A=.1
5460 IF B=0 B=.1
5500 X=0
5550 Y=0
5600 IF ABS(A/B)>4 X=A/ABS(A)
5700 IF ABS(B/A)>2 Y=B/ABS(B)
5800 IF (X=0) * (Y=0) X=A/ABS(A): Y=B/ABS(B)
5900 S=A(I)+Y*64+X*2
5910 REM S IS THE NEW LOCATION FOR ROBOT I
5915 PRINT AT A(I); "";
5920 IF A(A(I))<>.6 GOTO 6000
5940 REM ANOTHER ROBOT HAS ALREADY MOVED TO POSITION S
5945 REM SO DON't ERASE THE %
5950 FOR K=1 TO 30
5960 NEXT K
5970 PRINT AT A(I); "%";
5980 GOTO 6050
6000 REM MOVE ROBOT I
6010 A(A(I))=0
6050 IF A(S)=2 GOTO 30000
6100 IF A(S)<>-1 GOTO 6195
6110 REM ROBOT I HAS HIT A WALL AT S
6120 A(I)=0
6140 PRINT AT S; "%";
6160 FOR K=1 TO 30
6170 NEXT K
6180 PRINT AT S; "+";
6190 GOTO 8000
6195 IF A(S)=.6 A(S)=1
6200 IF A(S)<>.5 GOTO 6210
6201 REM ROBOT I IS MOVING TO POSITION OCCUPIED BY A ROBOT
6202 REM YET TO BE MOVED--.6 SIGNALS NOT TO ERASE %
6203 A(S)=.6
6204 A(I)=S
6205 PRINT AT S; "";
6206 FOR K=1 TO 30
6207 NEXT K
6208 PRINT AT S; "%";
6209 GOTO 6400
6210 IF A(S)<>1 GOTO 6300
6215 REM ROBOT I COLLIDED WITH ANOTHER ROBOT AT POSITION S
6220 A(S)=0
6230 A(I)=0
6240 V=V-1
6247 FOR J=1 TO 3
6250 PRINT AT S; "X";
6252 FOR K=1 TO 20
6253 NEXT K
6254 PRINT AT S; "";
6255 FOR K=1 TO 20
6256 NEXT K
6257 NEXT J
6258 REM FIND THE OTHER ROBOT THAT COLLIDED WITH I AT POSITION S
6260 FOR J=1 TO I
6270 IF A(J)=S A(J)=0: GOTO 6290
6280 NEXT J
6290 GOTO 8000
6300 PRINT AT S; "%";
6320 A(S)=1
6340 A(I)=S
6400 V=V+1
8000 NEXT I
9000 GOTO 2510
10000 REM SUBROUTINE TO PICK RANDOM LOCATION WITHIN ROOM FOR
10005 REM INITIAL POSITION OF * OR ROBOT OR WALL
10010 A=14
10015 B=42
10020 S=RND(B)+2
10030 S=INT(S/2+.5)*2
10040 S=RND(A)*64+S
10050 IF A(S)<>0 GOTO 10010
10060 RETURN
25000 REM OPPONENT WON
25100 PRINT AT 985; "YOU WIN!!! ";
25200 GOTO 170
30000 REM OPPONENT LOST
30010 FOR I=1 TO 5
30020 PRINT AT S; "*";
30030 FOR J=1 TO 30
30040 NEXT J
30050 PRINT AT S; "X";
30053 FOR J=1 TO 10
30055 NEXT J
30060 PRINT AT S; "";
30062 FOR J=1 TO 20
30064 NEXT J
30066 NEXT I
30068 PRINT AT S; "#";
30070 IF X=-3 PRINT AT 985; "SUICIDE ISN'T NOBLE! ";
30100 IF X<>-3 PRINT AT 985; "THEY GOT YOU! ";
30200 GOTO 170
```

Sample Run



(A slightly more difficult game — Difficulty of 10)

How to Fail with a Business System

BY RODNAY ZAKS

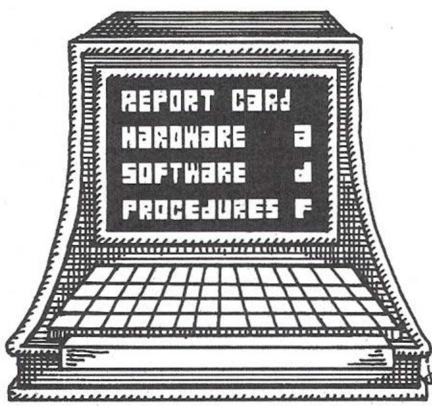
Failures here will be deemed to be "accidental;" i.e., due to errors. Every user of a business system should be aware of these errors in order to avoid them. A business system should be *secure and reliable.* Failure will be traced to three essential sources: hardware, software, procedures.

It is assumed that the system has been delivered in good operating condition. The first 100 to 200 hours of operation are called the *burn in* period. During this period, bad components are most likely to die. Many manufacturers or retailers will "burn in" a unit prior to shipment.

Mechanical failures are usually the most likely to occur in the life of the equipment and the best candidate is the printer. To obviate such failures, preventive maintenance is normally used in a production environment: mechanical parts are inspected, cleaned, and adjusted every x hours. This procedure catches most malfunctions before they occur.

All components of a system are rated within a specified temperature range and humidity level. Clearly, these specifications must be met. However, malfunctions due to temporary non-compliance are usually transient and will not damage the system permanently. As in any complex system, some of the electronic components might malfunction. General recommendation is to leave the diagnostic and repair to the supplier. This implies the availability of an efficient repair service locally.

If the system has been built according to generally accepted guidelines for reliable operation, hardware malfunctions are likely to be minimal after the first few days. They are normally entrusted to the manufacturer's repair service, or else to the local supplier. It is of critical importance for a business system that local on-site service be available.



Parity has long been a favorite technique for verifying correct transmission or retention of information. Parity consists in adding an extra bit to each byte of data to verify its contents. *Even* parity will add a "zero" if the total number of ones in the byte is even, and will add a "one" otherwise. In other words, it guarantees that the total number of bits will be even. *Odd* parity may also be used.

This technique will detect "single bit failures" which are the most likely to occur: if a single bit changes state, the parity verification module will compare the computed parity bit to the one stored with the byte, and detect the failure. Parity is extensively used in medium and large scale systems. It is almost never used in microprocessors. There are two reasons:

1 - Microprocessor systems are much more reliable than even traditional minicomputers, simply because they use far fewer components (the failure rate increases very quickly with the number of parts and interconnections)

2 - There has been no demand so far for the additional reliability at extra cost and complexity.

One area where a malfunction is likely to occur is in the memory of the system, because it uses a large number of components. A temporary failure during a space-war game is not objectionable: the game is just restarted. A temporary failure which wipes out hours of accounts-receivable data is certainly objectionable.

In addition, business systems are likely to require large RAM memory (40 or 48K bytes). Therefore, memory malfunctions are also more likely to occur. Eventually, there will be memory systems for microprocessors equipped with parity option, in order to provide the extra reliability.

Parity is used at the byte level. However, in the case of mass memories, (disk or tape), it is not possible to dedicate an extra bit to this function. In this case, a whole byte (or several bytes) are used at the end of specified blocks. This byte contains a "checksum" or a "CRC" (cyclic redundancy check).

The *checksum* is computed according to a simple formula involving the previous n bytes (where "n" depends on the system). If a byte is changed, the checksum is changed, and the checksum verification module will detect it when the block of data is read. CRC uses a more complex technique for computing the CRC bytes. Checksum or CRC are universally used in the case of disks or tapes. The checksum is

Reprinted with permission. From the book An Introduction To Personal And Business Computing *by Rodnay Zaks. Copyright © 1978, Sybex, Inc. The complete 245-page PB book, containing 14 related chapters on the computer's function in the business world is available for $6.95 from Sybex, Inc., 2020 Milvia, Berkeley, CA 94704.*

a simplified method, the CRC a more reliable one.

Finally, another good practice is the *"read after write"*: whenever a block of data is written, it should be read again. This is sometimes done in the case of disks, and should be used for crucial files. It is a software function. Usually it is not implemented as it slows down the write operation. Its availability as an option may be valuable.

Every complex program should be deemed incorrect! At least in the mathematical sense, there will almost always be some combination of events that will cause the program to malfunction. However, the probability is small, and, hopefully, little damage will result.

Because there is no way to guarantee that any long program, like any complex man-built system, is totally error free, every system will suffer occasionally from software bugs. Traditionally, manufacturers release periodic updates of programs where significant bugs have been found. For example, almost every BASIC interpreter, when first released, will misbehave when some specific sequence of instruction is used. These problems are then later corrected. Any software which has not yet been tested by a large number of users must be assumed to contain bugs. Any new software facility should, therefore, be treated cautiously in a business environment.

However, the point should not be overstated either: some users may never notice significant malfunctions. The system may not accept a command, and it will just have to be retyped. If good system design is used, the malfunctions are often limited to the need to repeat an operation which did not work. Even the largest IBM systems will experience catastrophic "crashes" on occasion. In fact, because of their very complexity, they are more likely to malfunction, and require a reliability enhancement technique.

Data should be verified when it is entered, and should also retain its accuracy when processed by the computer. *When entered, data must be checked for accuracy.* In many cases, this can be done by field checks and limit checks. A *field check* is also called a "reasonableness test." for example, a month should be between 1 and 12. An unusually large or small entry should be detected. A traditional computer horror story is the following: an inexperienced (or tired) typist in a government office enters a $1,000,000 order in the accounts receivable file, instead of $10,000. The manager has approved an expenditure rate of 30% of receivables. Before the error is caught, up to $300,000 could be spent.

Another possible horror story is the "Mr. Smith case." A $780.00 payment is received from Mr. Smith. It is promptly credited to his account, which showed only a balance of $180.00 due. A $600.00 refund check is issued. One month later, an irate Mr. Smith calls in, and asks why he is receiving a computerized "final notice" when he paid in full. You guessed why. There were two "Mr. Smith's" in the file. The payment was credited to the wrong one. Either a human, or else a software procedure should check for this situation. Ideally, the program will flag the "multiple Smith's" situation.

General business principles state that duties should be separated: the person verifying data should not be the one entering them. This reduces

the risk of accidental errors, and also introduces a control against malicious, or other modifications: Also, from a procedural standpoint: (and a vital business practice) *audit trails must be* correlated to a date, check number, and invoice number. Similarly, a commission list must be correlated to the transactions from which the commissions originated. In other items, sufficient cross-references must be listed to allow the complete verification of all data.

In situations where long codes are used, as with inventories, it is essential to sue either redundant encoding, or else check digits. *Redundant encoding* will use codes such as:

TUBE - 204211
BOLT - 418182
ASBL - 881921
FRME - 329137

The letters at the beginning of the code identify the product for a human operator. Every "tube" has a "204" code. Every "bolt" has a "418" code. The program will check that the code matches the letters. Errors will occur at most within the various tube or bolt types available; i.e. within the last 3 digits. This method is easy to implement, but wastes some space.

A *check digit* is an extra digit added to the code which will detect a transposition error, or a single digit error. For example: 881921 - 5. The "check digit" in this case is 5. It is computed by multiplying "1" by "1", then "2" by n^2, then "9" by n^2, etc., then adding up the 6 numbers. The sum is divided by n. If the remainder is R, the check digit is n - R. The choice of n varies. This method is almost always used when checking account numbers, as the most frequent error is a transposition, (writing 21224138 instead of 21221438.)

Files should be secure both in the case of machine malfunctions, and in the case of human error or interference. Naturally, all important files must be duplicated at regular intervals, and safeguarded. Just as importantly, software safeguards should be available, such as passwords or other access protection mechanism, from a good file system. A *password* can be imposed for a specific access to a file such as Read or Write. It must not be echoed. It must not be easily accessible within the system. Once created, a file can thus be protected against unauthorized access or modification by the wrong person. *File protection* features associated with a good file system are the ability to specify "access attributes" associated with a file, such as "write-protect". Finally, simple means such as a key on the terminal should not be underestimated.

The effects of computerizing can also be severe. Typically, unless operating personnel have been trained for the transition, significant processing delays may occur initially, resulting in degraded performance for a period of time. However, there is normally a quick transition to higher efficiency, provided that correct procedures are implemented.

Limited hardware and software failures must be expected. Provided that the precautions indicated be taken, these failures should not significantly affect the operation of the system. The greatest risk is usually in the procedures (programmed and managerial) used with the new system. Common sense, sound business judgement, and advice from an experienced user are the basic required resources. □

A General Game Playing Program

BY HERBERT L. DERSHEM

Programmers have used the look-ahead strategy to develop competitive game playing programs for games like checkers and chess. A general form of this look-ahead algorithm can be described in terms of a recursive procedure implemented in BASIC for specific games. If your BASIC processor accepts recursive subroutine calls, then you can use this algorithm to play any suitable game by programming three additional subroutines that describe the game. (For more information on recursive programming, see "Recursive Programming in BASIC", April *PC*.)

Consider a game with two players called "computer" and "opponent". At any given point in the game, two descriptors describe the situation: the game status (GS), often the status of the game board; and the player to move next (PM), either "computer" or "opponent". Each GS, PM pair results either in a completed game with a winner, or in a draw, or in a set of legal moves for PM. Each legal move maps the GS,PM pair. Let's consider the case where the players alternate moves, making the new PM generated by a move always different from the previous PM.

Now we're ready to recursively state the look-ahead algorithm which, given a GS,PM pair, evaluates all the legal moves available to player PM and determines the optimal one.

Algorithm Evaluate to find the best move BM for player PM from game status GS with evaluation of E.
Evaluate (GS,PM,E,BM)

1. If(GS,PM) is directly evaluatable, evaluate it and place result in E; return.
2. Generate MV_1,MV_2,...,MV_n, the set of all legal moves from (GS,PM), and GS_1,GS_2,...,GS_n, the corresponding set of game statuses after the legal moves are applied to GS.
3. If PM = computer, call Evaluate (GS_i, opponent, E_i, BM_i) for i = 1,2,..., n; for E_k, the largest of E_1,E_2,...,E_n, set E = E_k, BM = MV_k; return.
4. If PM = opponent, call Evaluate (GS_i, computer, E_i, BM_i) for i = 1, 2,..., n; for E_k, the smallest of E_1, E_2..., E_n, set E = E_k, BM = MV_k; return.

Evaluation of a game status is always from the computer's point of view. The larger the evaluation, the better the status is for the computer. Therefore, the principle behind this algorithm is that the computer always chooses from the legal moves that move resulting in a game status with largest evaluation. On the other hand, the opponent always chooses the move with the smallest evaluation, since that move is the least desirable for the computer.

How does the computer determine whether a move is directly evaluatable? If a game status is terminal, there are no further moves. Or sometimes the computer stops when a certain number of levels of moves have been examined. For example, a 3-level look-ahead will examine all of the computer's legal responses. As you can see, the number of moves that must be examined grows rapidly as the level of the search infinal level (level 3 in the example above), you must implement some heuristic procedure to evaluate the GS, PM pair. The ability of this procedure, the static evaluation function, to ac-

curately evaluate the game's status greatly affects how well the computer will compete. There's a trade-off between the depth of look-ahead and the validity of the static evaluation function. If the static evaluation function is perfect, the computer can use it to evaluate all its alternatives directly and not look ahead at all. On the other hand, if the computer can look ahead clear to the end of the game, examining all of the alternatives, it has no need for a static evaluation function since the perfect evaluation function is the game result: win, lose or draw. In practice we find ourselves somewhere between those two extremes.

For the general BASIC version for this algorithm, see Listing 1. Two additions to the algorithm have been made to speed up the search. Both halt the process when it's obvious no more searching is needed.

Suppose the search is at a level generating the computer's responses. If, at the preceding level, the opponent's best move evaluates to 4 and so far the computer's best move at this level evaluates to 5, why continue the search at this level? The opponent will never choose the current move under consideration because it will evaluate to no smaller than 5 which is already 1 worse than the best move the opponent has examined so far. This condition is tested in line 2100 of the program in Listing 1. In tree searching this process, called alpha-beta pruning, usually saves search time.

Additional savings can result from statement 2130 where, as soon as a player has found a sure winner for himself, he stops searching.

Now let's look at two implementations of the algorithm in Listing 1. The first, found in Listing 2, is the familiar game of tic-tac-toe. The implementation requires the addition of three subroutines to the general game status evaluator at 2000. These are 1000, a move generator; 3000, a static move evaluation function; and 4000, a game-over tester. But the choices shown here are examples: try designing your own improved versions of these subroutines.

The particular implementation here uses a maximum search depth of 10 levels. For tic-tac-toe, this level implies all searches will be terminated by the end of the game since the longest possible game is 9 moves. The static evaluation function returns 100 if the position is a win for the computer, -100 if it's a win for the opponent, and 0 if it's a draw.

Subroutine 2000 has been modified slightly from that shown in Listing 1 to accommodate the presence of only one subscripted variable in Radio Shack Level 1 BASIC, the system on which this program was implemented.

The ancient game of Kalah, our second game, is played on a board with six small pits on either side and large pits at each end. The game begins with 3 markers in each of the small pits as shown in Figure 1.

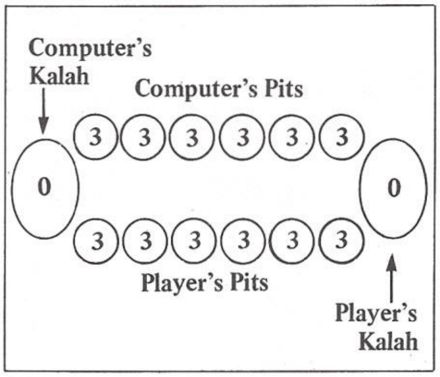


Figure 1 Initial position of Kalah board

The players alternate moves according to the following rules:

1. A player moves by choosing a pit on his side of the board and distributing the markers contained in that pit into other pits counterclockwise around the board beginning with the counterclockwise neighbor of the emptied pit. He places one marker in each pit and Kalah in turn until all markers removed are distributed. *Example:* If the opponent began play from the initial board shown above by emptying the fifth pit from the left on his side, after his move the board would look like Figure 2.

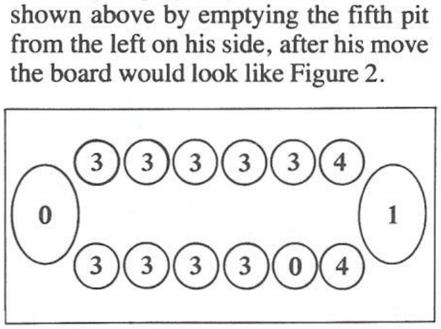


Figure 2 Example of a move

2. If the last marker distributed by a player lands in that player's Kalah, the player must empty another pit on his side. This move is called a continuation. The continuation might have another continuation, and so on. *Example:* If the computer's response to the above move was to empty the third pot from the left, it would receive a continuation. This move and its continuation are illustrated in Figure 3.

3. If the last marker distributed on a player's move lands in an empty pit on the player's side of the board, and if some markers are in his opponents pit directly opposite this pit, then the last marker distributed and all the markers in the opposite pit are placed in the Kalah of the player making the move. This move is called a capture. *Example:*If the opponent now empties the lands in the empty pit and captures the computer's four markers on the opposite side (Figure 4).

The winner has the most markers in his Kalah at the end of the game. When a player has no more markers in his pits and it's his turn to move, the game ends. At that point the opponent places all the markers in his pits into his Kalah, and the winner is determined.

Listing 3 shows the application of the game playing algorithm to this game. The continuation complicates matters by requiring two locations to store a move as well as a special coding scheme for continuation moves.

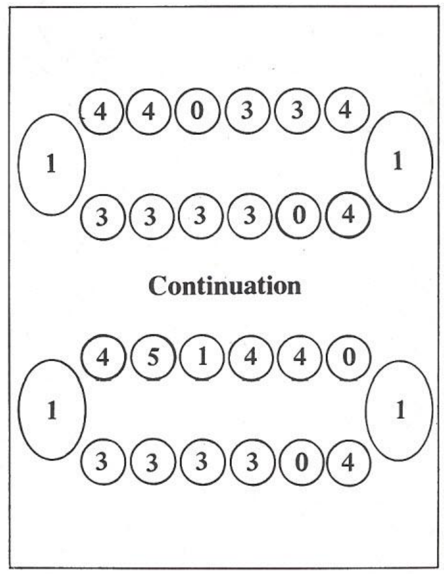


Figure 3 **Example of a move and continuation**

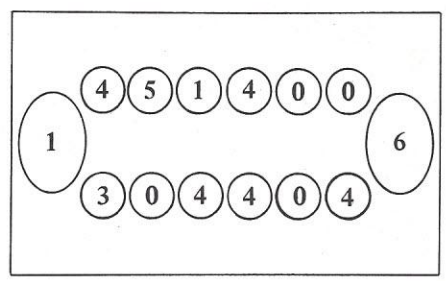


Figure 4 **Example of a capture**

Now that you've seen these examples, you can implement this algorithm for other games. You might want to improve the computer's performance on these games by providing better static evaluation functions or increasing the maximum depth of search. You must proceed with caution, however. Look-ahead algorithms can consume lots of computer time. So be prepared to wait for the computer's moves. □

Listing 1 – Game-Playing Algorithm

```
1500 REM GENERAL GAME PLAYING PROGRAM.  THIS SUBROUTINE,
1510 REM WHICH IS CALLED BY 1990, WILL ACCEPT A GAME STATUS STORED
1520 REM IN A(1) THRU A(S) AND RETURN IN S(1) THE BEST MOVE FOUND
1530 REM AND IN E THE EVALUATION OF THAT MOVE.
1540 REM PARAMETERS:
1550 REM     S IS THE NUMBER OF LOCATIONS NEEDED TO STORE GAME STATUS.
1560 REM     M IS THE MAXIMUM DEPTH OF SEARCH
1570 REM     N IS A VALUE WHICH IS IMPOSSIBLE CODE FOR A MOVE AND
1575 REM       REPRESENTS A NULL MOVE.
1580 REM     W IS A VALUE SUCH THAT ANY GAME STATUS WHICH EVALUATES
1590 REM       >=W IS A WIN FOR THE COMPUTER AND ANY WHICH EVALUATES
1600 REM       <=-W IS A WIN FOR THE OPPONENT.
1610 REM
1620 REM VARIABLES:
1630 REM     L IS THE LEVEL INDICATOR FOR THE CURRENT LEVEL OF SEARCH.
1640 REM     Z INDICATES PLAYER WHO IS MOVING: 1=COMPUTER,-1=OPPONENT.
1650 REM     Q IS THE STACK POINTER. IT INDICATES THE POSITION IN THE
1660 REM       STACK DIMENSIONED VARIABLE WHERE THE CURRENT GAME STATUS
1670 REM       DESCRIPTION BEGINS.
1680 REM     M(L) IS THE CURRENT MOVE BEING EXAMINED AT LEVEL L
1690 REM     S(L) IS THE BEST MOVE EVALUATED SO FAR AT LEVEL L.
1700 REM     B(L) IS THE EVALUATION OF THE BEST MOVE SO FAR AT LEVEL L.
1710 REM     E IS THE VARIABLE IN WHICH THE EVALUATION OF THE BEST MOVE
1720 REM       IS RETURNED.
1730 REM
1740 REM SUBROUTINES:
1750 REM 1000 GENERATES FROM MOVE M(L), THE NEXT MOVE IN A SEQUENCE
1760 REM    OF ALLOWABLE MOVES FROM THE GAME STATUS STORED AT POSITION
1770 REM    Q IN THE STACK. THE MOVE IS STORED IN M(L), AND THE NEW
1780 REM    GAME STATUS IS PLACED IN THE STACK BEGINNING AT POSITION
1790 REM    Q+S. THE FIRST MOVE IN THE SEQUENCE IS GENERATED WHEN
1800 REM    M(L)=N, THE NULL MOVE, WHEN THE SUBROUTINE IS CALLED.
1810 REM    IF M(L) IS THE LAST MOVE IN THE SEQUENCE, THEN M(L)=N
1815 REM    IS RETURNED.
1820 REM 3000 EVALUATES THE GAME STATUS STORED BEGINNING AT POSITION
1830 REM    Q OF THE STACK USING A STATIC EVALUATION FUNCTION. THE
1840 REM    VALUE IS STORED IN E.
1850 REM 4000 TESTS THE GAME STATUS STORED BEGINNING AT POSITION Q
1860 REM    OF THE STACK.  IF IT IS A GAME ENDING POSITION, THAT IS,
1870 REM    IF NO MORE MOVES ARE POSSIBLE, O IS RETURNED AS 1.
1880 REM    OTHERWISE, O IS RETURNED AS ZERO.
1890 REM
1900 REM THIS SUBROUTINE IS WRITTEN IN RADIO SHACK LEVEL I BASIC
1910 REM EXCEPT FOR THE USE OF EXTRA
     DIMENSIONED VARIABLES M,S, AND L.
1920 REM THESE HAVE BEEN USED FOR CLARITY.
1930 REM
1989 REM INITIALIZE L AND Z ON THE FIRST CALL.
1990 L=0: Z=-1
1999 REM UPDATE L,Q, AND Z FOR THE NEXT LEVEL OF SEARCH.
2000 L=L+1: Q=S*(L-1)+1: Z=-Z
2009 REM TEST IF GAME IS OVER.
2010 GOSUB 4000
2018 REM IF LEVEL IS TO THE MAXIMUM OR GAME IS OVER, EVALUATE
2019 REM USING STATIC EVALUATION FUNCTION AND RETURN.
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2049 REM INITIALIZE FOR BEST POSSIBLE MOVE SEARCH.
2050 M(L)=N: S(L)=N: B(L)=-Z*W
2059 REM GENERATE NEXT MOVE
2060 GOSUB 1000
2069 REM IF NO MORE MOVES, SET E AND RETURN.
2070 IF M(L)=N THEN E=B(L): GOTO 2150
2079 REM EVALUATE THIS MOVE, M(L), BY A RECURSIVE CALL.
2080 GOSUB 2000
2086 REM IF THE BEST MOVE AT THIS LEVEL IS ALREADY BETTER FOR Z
2087 REM THAN THE BEST MOVE FROM PRECEDING LEVEL WAS FOR -Z, THEN
2088 REM THIS MOVE WILL NOT BE CHOSEN BY -Z ANYWAY, SO RETURN
2089 REM WITHOUT EVALUATING THE OTHER MOVES AT THIS LEVEL.
2090 IF L=1 GOTO 2110
2100   IF Z*E>=Z*B(L-1) THEN B(L)=E: GOTO 2150
2108 REM IF THIS IS THE FIRST RESPONSE TRIED OR IT IS BETTER
2109 REM THAN THE BEST SO FAR, RECORD IT AS BEST SO FAR.
2110 IF (S(L)<>N)*(Z*E<=Z*B(L)) GOTO 2060
2120   B(L)=E: S(L)=M(L)
2129 REM IF THIS RESPONSE WINS, THERE IS NO NEED TO SEARCH MORE.
2130 IF Z*B(L)<W GOTO 2060
2149 REM ADJUST L,Q, AND Z AND RETURN.
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
```

Listing 2 – Tic-Tac-Toe

```
954 REM TIC-TAC-TOE MOVE EVALUATOR IN RADIO SHACK LEVEL I BASIC.
956 REM THE BOARD POSITION IS SORED IN A(Q) THRU A(Q+8) AS
958 REM               X              X
959 REM       A(Q)    X    A(Q+1)    X    A(Q+2)
960 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
961 REM      A(Q+3)   X    A(Q+4)    X    A(Q+5)
962 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
963 REM      A(Q+6)   X    A(Q+7)    X    A(Q+8)
964 REM               X              X
965 REM
968 REM A POSITION UNOCCUPIED CONTAINS A 0.
972 REM A POSITION OCCUPIED BY THE COMPUTER'S MARK CONTAINS A 1.
974 REM A POSITION OCCUPIED BY THE OPPONENT'S MARK CONTAINS A 4.
975 REM M(L), S(L), AND B(L) FROM THE GENERAL ALGORITHM (SEE FIG. 1)
976 REM   ARE STORED IN A(Q+9),A(Q+10), AND A(Q+11), RESPECTIVELY.
978 REM VALUES OF PARAMETERS:
980 REM    S=12
982 REM    M=10 (SEARCHES UNTIL COMPLETION OF GAME)
984 REM    N=0
986 REM    W=100
988 REM
990 REM
992 REM SUBROUTINE 1000 STORES IN A(Q+9) THE NEXT MOVE FOR BOARD
994 REM POSITION A(Q)-A(Q+8) FROM PREVIOUS MOVE A(Q+9). IF A(Q+9)=0,
996 REM FIRST MOVE IS RETURNED. IF THERE ARE NO MORE MOVES, A(Q+9)
998 REM IS RETURNE AS ZERO. NEW BOARD IS STORED IN A(Q+S)-A(Q+S+8)
1000 A(Q+9)=A(Q+9)+1
1010 IF A(Q+9)>9 THEN A(Q+9)=0: RETURN
1019 REM IF POSITION IS OCCUPIED, TRY THE NEXT ONE.
1020 IF A(Q+A(Q+9)-1)<>0 GOTO 1000
1030 FOR I=0 TO 8
1040   A(Q+S+I)=A(Q+I)
1050 NEXT I
1059 REM RECORD THE MOVE.
1060 A(Q+S+A(Q+9)-1)=(Z=1)+4*(Z=-1)
107C RETURN
1986 REM
1988 REM TIC-TAC-TOE VERSION OF GENERAL EVALUATION ALGORITHM
1990 L=0: Z=-1: S=12: M=10: N=0: W=100
2000 L=L+1:Q=S*(L-1)+1: Z=-Z
2010 GOSUB 4000
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2050 A(Q+9)=N: A(Q+10)=N: A(Q+11)=-Z*W
2060 GOSUB 1000
2070 IF A(Q+9)=N THEN E=A(Q+11) GOTO 2150
2080 GOSUB 2000
2090 IF L=1 GOTO 2110
2100 IF Z*E>=Z*A(Q-S+11) THEN A(Q+11)=E: GOTO 2150
2110 IF (A(Q+10)<>N)*(Z*E<=Z*A(Q+11)) GOTO 2060
2120 A(Q+11)=E: A(Q+10)=A(Q+9)
2130 IF Z*A(Q+11)<W GOTO 2060
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
2986 REM
2988 REM STATIC MOVE EVALUATOR FOR TIC-TAC-TOE.
2990 REM SUBROUTINE EXAMINES A(Q)-A(Q+8) AND RETURNS:
2992 REM    E=100 IF WINNING POSITION FOR THE COMPUTER.
2994 REM    E=-100 IF WINNING POSITION FOR THE OPPONENT.
2996 REM    E=0   IF DRAW POSITION.
2998 REM    E=-0.5 IF NOT A GAME-OVER POSITION.
3000 GOSUB 4000
3010 IF ABS(V)=100 THEN E=V: RETURN
3020 IF V=8 THEN E=0: RETURN
3030 E=-0.5: RETURN
3988 REM
3990 REM GAME-OVER TESTER FOR TIC-TAC-TOE.
3992 REM SUBROUTINE EXAMINES A(Q)-A(Q+8) AND RETURNS:
3994 REM    O=1, V=100 IF WINNING POSITION FOR COMPUTER.
3996 REM    O=1, V=-100 IF WINNING POSITION FOR OPPONENT
3998 REM    O=1, V=8   IF DRAW POSITION.
3999 REM    O=0, V<8   IF NOT A GAME ENDING POSITION.
4000 RESTORE: V=0
4010 FOR I=1 TO 8
4020   READ A,B,C
4030   T=A(Q+A)+A(Q+B)+A(Q+C)
4040   IF T=3 THEN V=100: O=1: RETURN
4050   IF T=12 THEN V=-100: O=1: RETURN
4060   IF (T=5)+(T=6)+(T=9) THEN V=V+1
4070 NEXT I
4080 O=(V=8)
4090 RETURN
4099 REM THIS STORES ALL 8 COMBINATIONS OF POSITIONS FOR WINNING.
4100 DATA 0,1,2,3,4,5,6,7,8,0,3,6,1,4,7,2,5,8,0,4,8,2,4,6
```

Listing 3 – Kalah

```
938 REM KALAH GAME EVALUATOR IN RADIO SHACK LEVEL I BASIC.
940 REM THE BOARD IS STORED IN A(Q) THRU A(Q+13) AS
942 REM
944 REM                          COMPUTER'S PITS
946 REM     A(Q+12)  A(Q+11)  A(Q+10)  A(Q+9)  A(Q+8)  A(Q+7)
948 REM A(Q+13)                                            A(Q+6)
950 REM     A(Q)     A(Q+1)   A(Q+2)   A(Q+3)  A(Q+4)  A(Q+5)
952 REM                          OPPONENT'S PITS
954 REM
956 REM A SIMPLE MOVE IS REPRESENTED BY AN INTEGER 0-5 WITH 0
958 REM   REPRESENTING THE PIT FARTHEST FROM THE PLAYER'S KALAH
960 REM   AND 5 THE PIT NEAREST THE PLAYER'S KALAH.
962 REM A CONTINUATION MOVE IS THE SEQUENCE OFI SIMPLE MOVES CODED
964 REM AS FOLLOWS:
965 REM    (MOVE1)*6(I-1) + (MOVE2)*6(I-2) +...+ (MOVEI)*6o.
966 REM THE LCCATION FOLLOWING THIS CONTAINS 6(I-1) TO INDICATE
968 REM THE NUMBER OF CONTINUATIONS.
970 REM M(L) IS STORED IN A(Q+14),A(Q+15)
972 REM S(L) IS STORED IN A(Q+16),A(Q+17)
974 REM B(L) IS STORED IN A(Q+18)
976 REM VALUE OF PARAMETERS:
978 REM    S=19
980 REM    M IS UNDER EXTERNAL PROGRAM CONTROL
982 REM    N=-1
984 REM    W=100
986 REM
988 REM SUBROUTINE 1000 STORES IN A(Q+14),A(Q+15)THE NEXT MOVE FOR
990 REM BOARD POSITION A(Q)-A(Q+13) FROM PREVIOUS MOVE A(Q+14),
992 REM A(Q+15).  IF A(Q+15)=-1, THE FIRST MOVE IS RETURNED.
994 REM IF THERE ARE NO MORE MOVES, A(Q+14) IS RETURNED AS -1.
996 REM THE RESULTING BOARD POSITION IS STORED IN A(Q+S)-A(Q+S+13).
997 REM
998 REM INCREMENT MOVE AND STORE IN T AND R.
1000 A(Q+14)=A(Q+14)+1: T=A(Q+14): R=A(Q+15)
1009 REM INITIALIZE NEW BOARD.
1010 FOR I=0 TO 13: A(Q+S+I)=A(Q+I): NEXT I
1019 REM IF MOVE AT ONE CONTINUATION EXHAUSTED,COME BACK A LEVEL.
1020 IF (INT(T/6)*6=T)*(R<>1) THEN T=T/6: R=R/6: GOTO 1020
1024 REM TEST FOR LAST MOVE.
1025 IF (T=6)*(R=1) THEN A(Q+14)=-1: RETURN
1030 U=T: V=R
1039 REM PULL OUT SIMPLE MOVE.
1040 X*INT(U/V): U=U-X*V: V=INT(V/6)
1049 REM P IS THE PIT POSITION ON BOARD OF MOVE.
1050 P=7*(Z=1)+X
1059 REM IF PIT IS EMPTY, GO GET ANOTHER MOVE.
1060 IF A(Q+P+S)=0 THEN T=T+1: GOTO 1010
1069 REM MAKE THE MOVE.
1070 D=Q+S: FOR I=P+1 TO P+A(D+P)
1080   J=I-INT(I/14)*14
1090   A(D+J)=A(D+J)+1
1100 NEXT I
1110 I=A(D+P): A(D+P)=0
1119 REM IF A NEW CONTINUATION IF SOUND, GO FORWARD A LEVEL.
1120 IF (J=6+7*(Z=1))*(V=0) THEN T=T*6: R=R*6: U=0: V=1
1129 REM MORE CONTINUATIONS?
1130 IF V>0 GOTO 1040
1139 REM TEST FOR CAPTURE.
1140 IF (A(D+J)<>1)+(P+I>=6+7*(Z=1)) GOTO 1170
1150   A(D+6+7*(Z=1))=A(D+J)+A(D+12-J)+A(D+6+7*(Z=1))
1160   A(D+J)=0: A(D+12-J)=0
1169 REM MOVE COMPLETED, SO RETURN.
1170 A(Q+14)=T:A(Q+15)=R:RETURN
1986 REM
1988 REM GENERAL EVALUATION ALGORITHM (FIG.1) FOR KALAH.
1990 L=0: Z=-1: S=19: N=-1: W=100
2000 L=L+1: Q=S*(L-1)+1: Z=-Z
2010 GOSUB 4000
2020 IF (L<=M)*(O=0) GOTO 2050
2030   GOSUB 3000
2040   GOTO 2150
2050 A(Q+14)=N:A(Q+15)=1:A(Q+16)=N:A(Q+17)=1:A(Q+18)=-Z*W
2060 GOSUB 1000
2070 IF A(Q+14)=N THEN E=A(Q+18): GOTO 2150
2080 GOSUB 2000
2090 IF L=1 GOTO 2110
2100 IF Z*E>=Z*A(Q-S+18) THEN A(Q+18)=E: GOTO 2150
2110 IF A(Q+16)<>N)*(Z*E<=Z*A(Q+18)) GOTO 2060
2120 A(Q+18)=E: A(Q+16)=A(Q+14): A(Q+17)=A(Q+15)
2130 IF Z*A(Q+18)<W GOTO 2060
2150 L=L-1: Q=Q-S: Z=-Z: RETURN
2982 REM
2984 REM STATIC MOVE EVALUATOR FOR KALAH.
2986 REM  E=100 IF WINNING POSITION FOR THE COMPUTER.
2988 REM  E=-100 IF WINNING POSITION FOR THE OPPONENT.
2987 REM ELSE:
2988 REM  E=((CONTENT OF COMP'S KALAH)-(CONTENTS OF OPP'S KALAH))*
2990 REM     (1 + 1/(19 - MAX CONTENTS OF A KALAH))
2996 REM
2998 REM WINNING POSITION?
3000 GOSUB 4000
3010 E=A(Q+13)-A(Q+6)
3020 IF O=1 THEN E=100*((E>0)-(E<0)): RETURN
3030 F=A(Q+13)*(E>0)+A(Q+6)*(E<0)
3040 E=E*(1+1/(19-F))
3050 RETURN
3992 REM
3994 REM GAME-OVER TESTER FOR KALAH.
3996 REM O=1 IF GAME IS OVER.
3998 REM O=0 IF GAME NOT OVER.
4000 O=0
4009 REM TEST FOR A WINNER.
4010 IF (A(Q+13)>18)+(A(Q+6)>18) THEN O=1: RETURN
4019 REM TEST FOR MOVER'S PITS ALL EMPTY.
4020 J=7*(Z=1)
4030 FOR I=J TO J+5
4040   IF A(Q+I)<>0 THEN RETURN
4050 NEXT I
4059 REM THEY ARE, SO EMPTY OTHER'S PITS INTO KALAH.
4060 FOR I=7-J TO 12-J
4070   A(Q+13-J)=A(Q+13-J)+A(Q+I): A(Q+I)=0
4080 NEXT I
4090 O=1: RETURN
```

Doubling Space on Single-Sided Disks

"The amount of data stored in a computer system expands to fit the space available." — Unknown

BY RODNEY L. WRIGHT

The above statement reflects a phenomenon that has perplexed data processors since the first installation of automated systems. As the number of applications increase, so does the need for more data and program storage space. As the amount of space increases, the number of feasible applications also increases.

Microcomputer users are not immune from such demands for ever increasing storage space. Low cost mass storage media such as cassette tape recorders were adapted to microcomputers to meet those demands. Cassette tape recorders have a serious disadvantage of inherently limiting the usefulness of microcomputer systems due to the large amount of time spent in input and output operations. The more recent development of reasonably priced mini-floppy disk systems was an attempt to provide large amounts of data and program storage on an immediately available "on-line" basis.

Once the mini-disk system is purchased, the microcomputer user faces the need to provide disk space on which programs and data can be recorded. In my case, I was so happy to be freed from the constraints of my previously tape-oriented system that I hadn't given much consideration to the cost of the disk media. It wasn't long, however, before the DISK FULL message reminded me that my data had exceeded the space available. Of course the solution was to buy more disks, and my library began to grow. Since I make backup copies of all disks, my requirements were automatically doubled. I soon realized that the $5.95 price per disk no longer appeared inexpensive.

Searching for alternatives, I found that other brands of disks provide equal quality at a lower cost and that bulk purchases reduce unit costs even further if at least ten disks are bought. Some vendors offer double-sided disks which sell at a premium of approximately $2 more than the single-sided variety. On a cost-per-byte basis, these dual-sided disks are less expensive than single-sided disks.

At a local personal computer show, one of the attendees suggested that I try to convert my single-sided disks into double-sided disks. Since my microcomputer is supported with a microbudget, the idea fascinated me and I decided to try it out. Comparing single-sided to double-sided disks, I could only determine the following differences:

- Double-sided disks have an extra set of timing holes
- Double-sided disks have an extra write protection notch

Single-sided disks convert easily to double-sided disks. Physical changes required can be accomplished quickly and with little or no expense. They only require a sharp knife and a 1/4" single, round hole punch. Here is the recommended procedure for conversion:

First, use a template to draw the new timing hole and new write protection notch on the plastic case of the mini-disk (see Figure 1). Make the template made from a stiff material such as the cardboard used for file dividers. Mark both sides of the plastic case. To protect the disk, it is a good idea to use a felt tip marker rather than a ball point pen or pencil.

Second, use a 1/4" round, single hole punch to create the new timing holes at the positions marked with the template. A small shield should be used to protect the disk surface against scratching while punching holes. Cut the shield from light card stock (such as a 3" by 5" card). Figure 2 shows a pattern for the shield which is slipped through the center hole and placed between the disk surface and the plastic case while punching the new timing holes. Make sure that your punch is not magnetized before trying this change! I found the conductor's style of punch is most useful in this operation.

Third, cut out the write protection notch at the position marked with the template. A hobby knife or carver's tool works well, but any sharp bladed knife would suffice. An even quicker and neater notch is made with the "nibbler" tool available at electronics stores.

Finally, use the disk operating system's backup utility program to format and verify the newly available disk space.

Of my twenty disks, only one is not usable on the reverse side. (The backup utility rejected the disk because it had a bad track.) Thus I was able to nearly double my disk space for an investment in a single hole punch — which costs less than a single disk! □

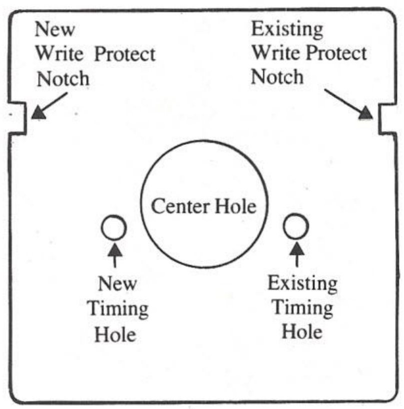


FIGURE 1. Double-sided Disk Template

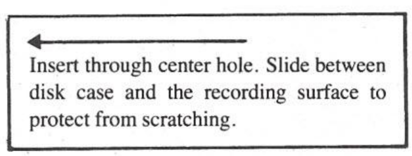


FIGURE 2. Disk Surface Protector

HARRY SHERSHOW – Dept. Editor
MORRIS MILLER – Chess Annotater

A Russian Algorithm for Chess

(The Russian chess program, KAISSA, named after "Caissa" the mythological goddess of chess, emerged on the world chess scene in 1974 by winning the First World Championship computer chess tournament in Stockholm. KAISSA was developed by a team of Russian researchers headed by V. Arlazarov, of the Institute of Management Problems, Moscow. Another Russian program, which is currently being improved, is PIONEER. It was designed to solve end games, and was developed by famed chess savant, Mikhail Botvinnik, who heads a laboratory at the Moscow Institute of Power Engineering. The other members of the PIONEER team are A. Yudin, B. Shtilman, D. Lozinsky, and L. Panfilova. The January 1979 issue of PERSONAL COMPUTING showed the PIONEER program solving an end game problem. The following article "An Algorithm for Chess", is reprinted from the "Soviet Union Review". It was written by Russian chess master, V. Khenkin, who analyzes PIONEER in the process of solving another type of end game problem.)

"Theoreticians in many countries are engaged in intensive research into chess-playing computer programmes," writes Khenkin. "The question may arise: is it necessary to spend all this energy, time and money to build a chess robot? The development of a chess computer is not an aim in itself. Chess is a convenient model for scientists in quest of knowledge of the human mind in action. The problems the player has to cope with in a game of chess are akin to a broad range of problems of immense importance for humanity. 'Management science is developing apace,' says Academician V. Trapeznikov, director of the Institute of Management Problems. 'It's multiple-tiered and covers not only, say, the running of ships, aircraft and technological plants but also the management of large groups of people. . . Chess is indeed an ideal model task, and its solution noticeably facilitates a comparison of management action—a game can be replayed but two factories cannot be built to see which project best fits in with the overall plan.'

"Academician N. Yakenko said: 'The selection of an optimal strategy is one of the most complex intellectual tasks. Yet it is the questions of strategy that humanity has now to deal with. In particular, man has to upgrade his long-range economic development strategy and environmental strategy in order to both preserve nature and make wide use of its wealth. To tackle these problems scientifically we must present proof of our abilities. In this context a game of chess is a qualified test for the computer. . . If we cannot learn how to cope with strategic problems in chess we will hardly be able to shoulder more complex tasks.'

"It is generally recognised that the idea of a chess robot was conceived in 1949 by the American scientist Claude Shannon. He had a rather hazy idea of chess and evidently had no inkling of the torments to which he was subjecting his colleagues. The father of cybernetics, Norbert Wiener, was very optimistic about chess-playing computer programs. He said that in 25 years a robot would play like a master. The 25 years have passed but the robot is still a novice. The hitch is that neither chess players nor theoreticians can explain the chess-playing process. In terms of pure theory the game is based on an analysis of variations. Modern computers operate at a speed beyond man's abilitities. This being the case, is it possible for a robot to play a game from beginning to end in one go?

"A chess magazine once carried a cartoon showing a computer playing chess with a professor who opens with P-K4; the computer resigns because it calculates that the Black could be mated on the 999th move. The computer could do this if it were able to sort out the 10^{120} number of possible variations in one game. Do not try to decipher how much 10^{120} is. Specialists believe that it by far exceeds the number of atoms in the Universe.

"Claude Shannon worked out two ways of processing chess information: (1) trying all the possible moves; and (2) trying those moves relevant to the ultimate goal of the game. Naturally, the second method is more productive and it is used by players. The human player can limit the number of variations to be analysed and can single out priority moves. The computer, on the other hand, is unaware of the reasons guiding the player. This appears to be the key problem and makes the computer a learner to this day. Its solution should instill character into the robot's style of play.

"A capacity audience came to the main auditorium of the USSR Central Chess Club in Moscow one evening in December 1968 to hear Mikhail Botvinnik deliver a lecture on his research into chess programming.

"'Only theoreticians have tackled the problem of a chess robot so far,' he said. 'They have tried to devise a machine which would not play the same way a human being does. They invented man-made ways of chess playing and transferred them to their punch-cards. Despite some headway, this approach could not bring significant progress. Why did theoreticians fail to develop a strong robot player? The fact is that all their programmes were based on the method of the so-called complete tree of variations. It means this. Let us assume that when White considers his first move he should anticipate all of the, let us say, 30 possible countermoves by Black (figuratively speaking, 30 branches shoot off the trunk). Each successive move sets off another 30 branches, and so on almost ad infinitum. To choose the best move the computer would have to consider an infinite number of variations.

'Yet a chess player knows that he never examines all the possible moves. He singles out the main ones. To teach a chess robot to play like a master it is therefore necessary to find an entirely new approach to this problem, an approach different from the one which mathematicians have used so far. It is necessary, in the first place, to formulate the method used by chess players and then pass it on to the robot. However, the mathematicians did not and could not know, of course, how a grandmaster does it at the chessboard. I can suggest something new.

'We all know that the purpose of the game is to capture the opponent's King. Yet on the way to this overriding aim a player pursues intermediate objectives, or the objectives of **inaccurate erratic play,** as we might call it. I made the seemingly very simple assumption that the object of **inaccurate play** in chess is aimed at winning pieces. This should not, of course, be simplified to mean winning a piece at once. The way has to be paved for winning a piece in the future; that is, a favourable outlook in terms of captures has to be created. If my assumption is correct, then everything in chess begins with an attack.'

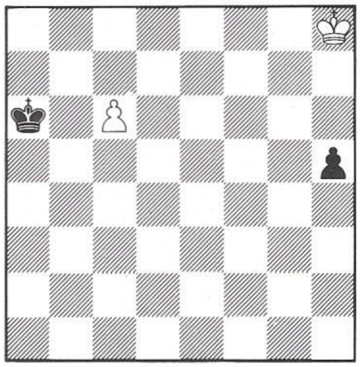
"In short, Botvinnik suggested that attention should be paid only to those variations which lead, or could lead, to material gains. To begin with, **trajectories of attack** are calculated for each chess piece. To shorten the range of attacks which reach too far, the length of trajectories is limited by a certain number of semi-variations, or a **horizon** to be shifted depending on how complicated the situation is. The **trajectories of attack** take into account opposition by the adversary and support by one's own pieces. All these factors make up a **play zone** and serve as a basis for drawing up a **tree of variations.** According to Botvinnik's algorithm, computer-chess playing becomes intelligent and purposeful, and the number of moves to be considered in choosing the best one is sharply reduced compared with other programmes.

"Botvinnik's idea met with scepticism at first. Ex-world champion Mikhail Tahl said caustically: 'It's Botvinnik who thinks that this is the way he thinks.' Many theoreticians questioned the possibility of using his method at all.

"Meanwhile Botvinnik and a team of young colleagues continued their intensive research at the Institute of Power Engineering. They also had sponsorship from the Computer Centre of the USSR State Planning Committee. At 8 p.m. Moscow time, on January 28, 1977, the Pioneer programme based on Botvinnik's algorithm showed its worth on the chessboard for the first time by solving an end-game study compiled by the Czechoslovak Grandmaster Richard Reti in 1921 (see diagram). White to play and draw.

"At first glance the problem seems to be absurd: the White King is positioned far behind the Black Pawn on the R4 square, whereas the Black King can easily block the White Pawn on the QB6 square. Yet a draw is possible. White maneuvres his King in such a way as to catch up with his opponent's Pawn while threatening to support the advance of his own Pawn.

The Reti Problem (1921)
White to play and draw



"**1. K-Kt7, P-R5 2. K-B6.** A curious situation. If Black moves the Pawn to R6 White will swing the King in the other direction. 3. K-K6, and Black will be powerless to stop the White Pawn. For instance, 3. ... P-R7 4. P-B7, K-Kt2 5. K-Q7 or 3. ... K-Kt3 4. K-Q6, P-R7 5. P-B7, K-Kt2, and two Queens go into action one after another.

"**2. ...K-Kt3 3. K-K5.** As in the above variation, White threatens 4. K-Q6 in support of his Pawn.

"**3. ... KxP 4. K-B4** and there can be no doubt here that the White King will catch up with and capture the Black Pawn.

"How does the Pioneer programme go about it? First, it plots **trajectories of attack** with a preset horizon of ten semi-variations. Priority goes to trajectories aimed at major targets, the Kings in the given case. For White this is a trajectory of the Pawn: B6-B7-B8-Queen x P, and for Black it is the trajectory of the Pawn: R5-R4-R3-R2-R1-Queen x P. The machine tries a straightforward play: 1. P-B7, P-R5 2. P-B8-Queen, and learns that Black will fall behind. The pseudo-variation is erased from the computer's memory, and the system is returned to its initial position. Now it follows the **trajectories of control** over the B8 square (see the **tree of variations** diagram) by the White King.

"Before we follow the robot's rea-

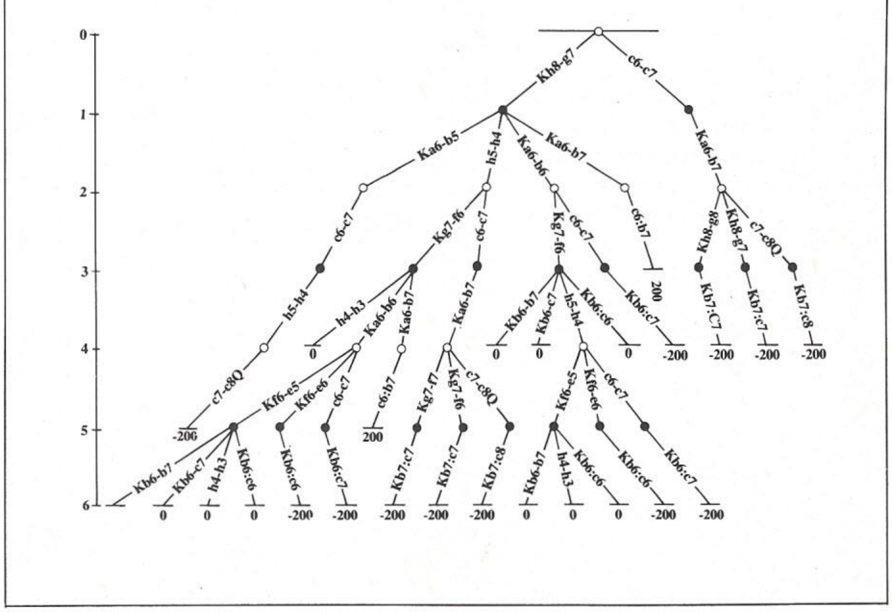


Diagram of "tree of variations" as compiled by the PIONEER program in solving the Reti problem. The outlined rings are White's moves; the solid circles are Black moves. Vertical bars are ends of variation.

soning let us decipher some notations on the **tree of variations** diagram. The figures at the end of each variation stand for the assessment of a situation in terms of material. 'O' is balance, and this is the goal — a draw — that White is after. All negative numbers mean an advantage for Black, trying to avoid an equal position. The robot is without bias and plays with equal vigour for White and Black. Both White and Black change variations if the outlook is negative. The robot goes back to the initial position, moving from variation to variation until all the programmed variations are tried.

"Let us go back to the end-game study. The machine moves 1. P-B7, and the programme selects a **trajectory of negation** K-Kt2 xP. The variations show that after 1. ... K-Kt2 2. P-B8-Queen the balance is tipped in Black's favour. This sets off the **trajectory of control** over the B8 square by the White King, and the machine changes White's latest move. The variations 2. K-Kt7, KxP and 2. K-Kt8, KxP again put White to a disadvantage, and the game is played from the very beginning.

"The robot now starts with 1. K-Kt7. Not with 1. K-Kt8 because the White King is not only after the Black King, its chief adversary, but also after the Black Pawn. This is an example of a **two-pronged variation** (to use Botvinnik's words), which has received preference in the programme.

"The robot tries the variations in the two **play zones:** one for the White Pawn and the other for the Black Pawn. The reader can trace the variations and assessments of the situation on the map diagram. After all the variations are tried, as demanded by the program, the game starts anew. The robot now has every reason to offer 1. K-Kt7 as the best variation. It is marked 'O,' or balance. The robot considers Black's moves KxP and P-R6 and justly terms them as equalizers. This is true because they are countered by White's K-B4 and K-Q6, and theoretically the game should be drawn according to the so-called notion of the square. The robot has 14,000 positions and notions of end-game techniques stored in its memory. After each move it considers all the references in its huge collections.

"The Pioneer program contains only 54 groups, while a full-scale program would have to examine at least 1,000,000 moves. The advantages of this contracted and target-oriented method are fully evident. It took a computer operating at a speed of 300,000 operations per second 68 minutes and 5 seconds to cope with the problem. A computer working at 10,000,000 operations per second could do it in 3 minutes, or about as fast as a top-class master.

"'We are now checking the mechanism of the whole programme without positional assessments,' said Botvinnik. 'Solution of end-game studies serves this purpose, for they do not have long-range positional evaluations. The variations are all very much to the point; the search for the best moves is conducted along a wide front and in original situations. The initial results showed us to be on the right road. Before long we will introduce positional assessments of a general type into Pioneer and it will be able to play a game of chess from beginning to end. This will probably happen this year. **I expect Pioneer to play approximately at master level.**

"Ex-world champion Dr. Max Euwe once said: 'Botvinnik thinks a machine will soon play like a master but I think it will take at least a century to do this.' Recall, however, that Jules Verne predicted man would not be able to transmit images across distances before the 29th century, and that Lord Rutherford maintained that man would never be able to use the energy of the atomic nucleus.

"It is hard to say today in which fields of knowledge Botvinnik's methods can be applied. In his preface to the American edition of a book on Botvinnik's work Arthur Brown said that 'An Algorithm for Chess' is a contribution to science at three levels: at the immediate level, it provides a basis for a computer program that seems likely to succeed in playing chess; at the middle level, game-playing programs help us to study and rationalize the processes of planning and decision-making; and, at the highest level, the study of the mind in action, as in the game of chess, leads to an understanding of human thought and of the human psyche."

The Longest Game

Don Gerue reports on the latest developments in the tournament that he and Russ McNeil are directing:

"Phase Two of the Penrod Memorial Tournament has been ready to start for the last month except that the Atari Program is still in the preparation phase and has not been available for the Tournament. Compuchess II is on hand, as is the JS&A $100.00 Chess Computer. When Atari arrives we will get the Tournament underway again.

"One of the best things that has happened to me to date, as a result of the Tournament, has been the opportunity to help Dan & Kathy Spracklen get SARGON II ready for the 9th North American Computer Chess Championships. To work with these two top notch programmers as they corrected the last few bugs in the program prior to leaving for Washington, last December provided more insight into chess programming than any books available.

"As everyone has heard by this time, SARGON II tied for third place in the Championships, clearly beaten by only BELL and CHESS 4.7. Who ever said that only big computer would ever play a really top notch game of chess? Whoever it was, must have a full stomach by now from eating his words. From recent conversations with Dan and Kathy I can tell you that improvements are still in process. Using some of the same techniques proven by CHESS 4.7 SARGON will have iterative deepening and a greater choice of operating levels. While SARGON can not move as quickly on the TRS 80 as it can on the Wavemate Jupiter (1.77 vs 4.25 MHz clock) it should come close to doing as well as it did at the ACM Tournament within the time of three minutes."

During the first Penrod Memorial Chess Tourney in Santa Barbara, which took place last year, the longest game of the tournament occurred between

Microchess 2.0 and BORIS. "It took almost 6 hours," recalls Don Gerue, co-director of the tournament. "And when it was over, Russ McNeil — the other director — and myself could barely stay awake. Didn't get to bed until 4 o'clock in the morning. It was a good thing because the next day was Sunday and I didn't have to get up."

MICROCHESS 2.0, produced by Peter Jennings' Microware, Ltd., was one the the first microcomputer chess programs to appear on the market. It was undoubtably influenced by the big computer-chess programs that were becoming popular several years ago.

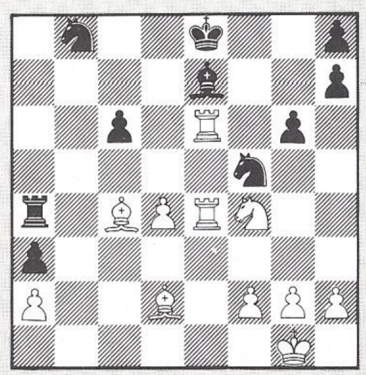
BORIS, on the other hand, is a comparative newcomer. It appeared shortly after CHESS CHALLENGER made its debut, and the two have been competing against each other in the growing market ever since.

What's new with BORIS these days? Dan Neumayer, of the Chafitz Com-

White — *MICROCHESS 2.0* Black — *BORIS*

1. P-K4	P-Q4
2. B-N5ch?	P-QB3
3. B-R4	PxP
4. Q-K2	P-KB4 (a)
5. Q-R5ch	P-KN3 (b)
6. Q-K2	P-K3
7. N-QB3	P-QN4?
8. B-N3	P-QN5?
9. N-Q	Q-Q2?
10. Q-K3	P-K4
11. N-K2	Q-K2
12. O-O	P-QR4
13. R-K	P-QR5
14. B-B4	P-KB5?
15. QxKP	B-B4
16. Q-B3	BxP
17. NxP	P-K5
18. Q-K3	BxN
19. RxB	N-KB3
20. R-K	N-N5?
21. QxP	N-KB3
22. QxQch	BxQ
23. R-K6	R-R2
24. P-Q4	P-R6
25. PxP	PxP
26. B-K3	N-N5
27. B-Q2	N-KR3
28. R(R1)-K	N-B4
29. R(K1)-K4	R-R5?
30. B-N3(c)	R-R2(d)
31. R(K6)-K5(e)	R-N2
32. R-K6	R-R2
33. P-N4	N-N2
34. R(K6)-K5	R-N2
35. R-K3	N-R3
36. B-B4 (f)	N-N
37. RxP	R-N8ch
38. B-K?	R-B8
39. R-QB3	RxR
40. BxR	R-B
41. R-K4	K-Q2
42. N-K6	NxN
43. BxNch	K-B2
44. B-R5ch	K-N2
45. B-B8ch	KxB
46. RxB	P-KR4
47. R-B7ch	K-Q
48. RxPch	K-Q2
49. R-B7ch	K-Q3
50. B-N4ch	KxR
51. BxR	PxP
52. K-N2	K-Q
53. P-Q5	N-R3
54. P-Q6	N-B4
55. B-K7ch	K-Q2
56. K-B	N-K5
57. K-K	NxQP
58. B-B8?	N-K5
59. P-QR3	N-N4
60. K-K2	N-B6
61. P-R3?	N-N8ch
62. K-B	NxP

White — *Microchess 2.0* Black — *BORIS*



Position after Black's 29th move (R-R5)

63. P-R4	N-B5
64. B-R6	N-K3
65. K-N2	N-B4
66. P-R5	K-Q3
67. B-B4ch (g)	K-Q4
68. K-B	N-Q6
69. B-K3	N-B4
70. K-K2	N-R3
71. K-Q2	K-K5
72. K-B3	P-N6
73. K-B4	P-N7
74. B-Q4	K-B6
75. K-B3	P-N8=Q
76. B-N6	Q-K8ch
77. K-B2	Q-K4
78. K-Q	N-B4
79. BxN	QxN
80. P-R5	KxP
81. K-Q2	Q-Q5ch
82. K-B2	K-N6
83. K-N3	K-B5
84. K-R2	Q-B5ch
85. K-R3	QxPch
86. K-N4	Q-Q3ch
87. K-B3	Q-B3ch
88. K-Q4	K-B4
89. K-K3	Q-Q4
90. K-B2	Q-Q6
91. K-N2	K-B3
92. K-R2	Q-K6
93. K-N2	Q-Q6
94. K-B2	K-K2
95. K-K	P-N4
96. K-B2	P-N5
97. K-N2	K-B3
98. K-R2	K-K3
99. K-N	K-B3
100. K-B2	P-N6ch
101. K-N2	Q-QB6
102. K-R3	K-K3
103. K-N4	P-N7
104. K-N5	P-N8=Qch
105. K-B4	Q(B6)-K6 mate

Annotations by Morris Miller

(a) Better is 4-. . .N-KB 3: 5-N-QB3, B-B4, etc.

(b) Now black's queen bishop has no scope.

(c) Now MICROCHESS has a winning line: 30-P-KN4, RxB; 31-PxN followed by RxB. SEE DIAGRAM.

(d) Avoiding 30-. . .RxP?; 31-RxBch, NxR; 32-RxR, etc.

(e) Now MICROCHESS overlooks 31-P-Q5! and if N-N2, then 32-P-Q6, winning.

(f) Again P-Q5! wins.

(g) Why not B-B8ch, followed by BxN and K-N3, etc.?

pany, Rockville, MD., has divulged the latest information on the growth progress of BORIS:

"First of all," says Dan, "we have two new products that we showed at the January Las Vegas Consumer Electronics Show. Both should be on the market around the middle of May. The first product is BORIS DIPLOMAT. It operates on disposable penlight cells so you can carry the game around with you wherever you go. It also comes with an adapter so that it can be run on house current. It has every feature BORIS has except those 'talking' messages. But it has the same 40% increased playing strength of last year's BORIS and shows the same rank display. We use different symbols for the DIPLOMAT because we don't have an alpha numeric display. It's not elaborate. Just a 7-segment display where the chess symbol is made up of combinations of seven fractional lines of the LED readout. This unit sells for $119.95. It has the same chess playing program and rating as previous BORIS' In fact, we've given BORIS to some Chess Masters and Experts and they all seem to agree that BORIS would be rated at about 1500 if you gave the unit a half hour to move. We know it's a long time-interval but we've found there are people who don't mind the time factor: invalids, shut-ins, prison-inmates, and so forth. And for other people, they can play intermittently. They go about their regular business and keep coming back to play the game as it progresses. They don't have to sit around and wait for every chess move. So, for some people, the time factor is really not that disastrous.

"Our other new product is called BORIS GRANDMASTER. The basic idea behind GRANDMASTER is to display the whole chessboard. The display is about 12 inches square — it has large red LEDs with 16 segments — in place of the seven — and the symbols are quite nice. The Knight, for example, looks like a horse. Next to the display is the BORIS Chess Computer, in the same box. You operate it the same way as you would the regular BORIS. There's a new feature now — a switch on the unit that is labeled 'Scan'. With the switch at 'Off' what you see are the actual pieces on the chessboard and you can watch as they make their moves during the active game. When you flip to 'Scan' you can observe BORIS moving through its search tree. It's like watching an actual action picture of the thought process as BORIS moves the pieces around on the chessboard while it is thinking. For example, in a three-ply search, you can easily follow the move on the first play (White's first possible move, for instance.) You can also follow the second play (Black's possible response.) The third play (White's next move) is, however, going to be happening so fast, as BORIS flies through thousands of 'test moves'. that all you're going to see is a blur. BORIS is now in its complicated, lengthy, tree-search. But still, in looking at it, you can see the order in which BORIS tests different moves. For example, after he finishes the two-ply search you can see him start his three-ply search with what he considers the best move of the second ply. After exhausting the search for that move, he goes back and tests the second-best move and starts another search. He'll examine all the second-ply moves in the same fashion, and you can see him do this quite clearly. It's like watching the thinking process in action.

"The GRANDMASTER has rechargeable batteries. The electric display board, however, draws too much current. So when BORIS operates on batteries alone, the electronic board doesn't work. There's a separate transformer plug-in to run the board. But, without the board display, you can follow the game as before because BORIS, on the small computer-unit, prints out the moves on the LED display and will show the chessboard by stepping up through the ranks as before. With the big chessboard display, you can set up special positions, end-games and so forth. The GRANDMASTER has memory-location features also. After you shut off the board the unit will retain location of all the pieces on the board for as long as a week. So, if you want to take the game next door for example, or go on a trip someplace today and finish the game tomorrow, you can turn off BORIS and continue to play later. It will retain the memory for a week without recharging. If you recharge the batteries every two or three days, you can keep the game in memory forever. To guard against weakening batteries, which might affect the memory, we have controls in the circuit which take care of current fluctuations. GRANDMASTER was developed as result of questionnaires we sent out to a lot of people. They told us what they wanted and we incorporated their ideas into the model.

"The GRANDMASTER will sell for $995. Now, Chafitz realizes that that is a lot of money and people wouldn't be happy with a model that stayed at a fairly constant level. Chess machines, like human chess players, should improve in the passing of time. So, that's what we're planning to do with GRANDMASTER. We are working out the details now. If anybody is going to spend $1000 for a chess computer they will want to be sure that it will always be the best chess computer they can get. Owners will be able to send their units back to us — we haven't worked out all details, yet — and because we use masked proms, we can upgrade it to the best available level at that point and return it. With David Slate and Larry Atkins still chugging away at our program, BORIS should become the best chess computer on the market this year and year after year afterwards."

The "marathon" game at the Penrod Memorial Chess Tournament is documented nearby, together with Morris Miller's analysis. It is remarkable that two microcomputer games could play for so long a period of time without reaching a draw or stalemate. It is true that both programs lost their queens by the 22nd move, a condition that usually leads to long games. But on the 75th move, BORIS was able to promote a Pawn to another Queen at which point the game should have come to an early conclusion. But MICROCHESS 2.0 went on for another 30 moves avoiding its inevitable defeat, when, at the 104th move, BORIS promoted a Pawn for a second Queen: a plucky display by one program to delay its defeat and a gritty determination of the other program to capture its elusive target. It was an interesting computerized example of the cat and mouse game, a Grandmaster watching one computer chase the other around the board for six hours might have delivered a karate punch in exasperation, splitting the board in two, and crying "Enough!"

"X" marks the Bishop

. . . The missing piece on White's KR4

in the game position shown last month, is White's Black-square Bishop. The logic in determining the missing piece appeared in Alan Gottlieb's Puzzle Corner, a Monthly Feature of MIT's Technology Review. The solution was submitted by several of Gottlieb's Puzzle-corner followers. "First," explained the puzzle solver, "Black's King is in check, so White's last move was to take Black's piece on Q8 with his pawn, promoting it to Rook. The piece taken by this Pawn was not a Rook or a Queen, since White's King is on the 8th rank and White would have been in check prior to Black's last move. Black's black-square Bishop must never have moved (Pawns are on Black's K2 and KN2) and there are two Black Knights on the board, so the piece taken by this pawn was a promoted Black pawn. This accounts for all eight black Pawns. The piece could not be Black's Queen or a Black Rook, again due to the placement of the White King. Also, Black's dark-square Bishop and his two Knights are accounted for, as are all his pawns as explained before. Thus the piece was White. The minimum number of White pieces taken by Black's Pawns (including the KRP which promoted) is 5. White has 10 pieces shown on the board, so 5 is the maximum number of pieces Black can have taken in order for a piece to be on White's KR4. All five pieces taken by Black's pawns were on White's squares, so the only possible piece left is White's black-square bishop.

Chess Challenge

. . . An item which should stir the gaming instincts of many computer chess programmers, appeared in the February issue of the ICCA newsletter: "A Dutch firm is offering a prize of $50,000 to anyone who programs a computer to beat Prof. Max Euwe, former president of the International Chess Federation, at his own game. The program must be drawn up during 1979 and the contest will consist of four matches. Amsterdam-born Euwe, now 77, was a world champion in the 1930s. He is also a computer specialist. Volmac, the automation company which sponsors the Rotterdam Chess Club, believes that grandmasters will never be beaten by computer programs."

Detroit next

. . . The 10th North American Computer Chess Championship, to be run in conjunction with ACM-79, will take place in Detroit, MI, Oct. 29-31. Organizing the tournament (as usual) will be Prof. M. Newborn, School of Computer Science, McGill University, Montreal, Que., Canada; and Prof. B. Mittman, Vogelback Computing Center, Northwestern University, Evanston, IL 60201. In charge of local Detroit arrangements will be David M. Dahm, Burroughs Corporation, Burroughs Place, Detroit, MI 48232.

More on Pillsbury

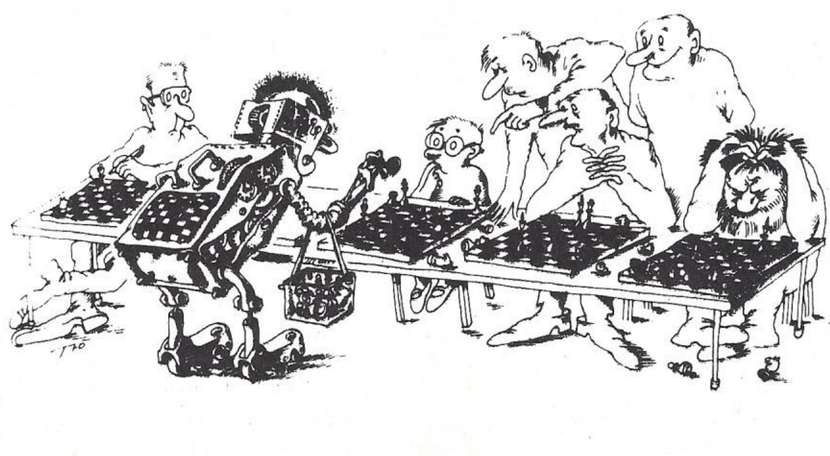
. . . Harry Lyman, of the Boylston Chess Club, Boston, writes to clear up a vague statement that appeared here on Pillsbury. "Pillsbury took up chess at age 16," writes Harry. "So he was at the Boston YMCU somewhere around 1886. He played Steinitz in April 1892 and won a short match 2 to 1. (Steinitz had offered a pawn and move odds). In April of 1893, Pillsbury, then only 21, won a match in Boston from Walbroot and later from Schottlander. Later in that year, he launched his serious chess career by playing in Philadelphia. In 1894 he competed in New York and then went to the world match at Hasting in 1895. I must clarify my stand on BORIS. My current opinion is that BORIS is unable to improve a beginner's skill as a chess player except under those conditions where the beginner can learn from BORIS's errors and try to correct them. BORIS's chess techniques have not been in evidence consistently enough to aid the newcomer — especially in the middle and end game. But in one direction BORIS is unequalled: a checkmate in one is spotted practically instantly."

Some kit, this BORIS!

. . . The Chafitz Co., 1055 First Street, Rockville, MD, is now offering its chess-playing BORIS in kit form. "You don't have to be an electronic expert to assemble this masterpiece," says the company. "For $175 you get a complete set of components with easy instructions. Blueprints are also included for building your own enclosure. Any chess player should have a lot of fun building this device, then have more fun trying to beat it at chess."

New time clock

. . . "Isn't it strange that mechanical chess clocks, which are still produced to specifications patented in the year 1900, are used to time games between computers?" That is a statement contained in a letter from Joe Meshi, of Micro General Corporation, PO Box 17746, Irvine, CA 92714. His company makes a digital chess clock, the "Micromate-180" which features a totally programmable time control, display of time to the nearest second and many functions which are new to the timing of chess games, according to Meshi. The clock is controlled by the CMOS based microprocessor and offers 20 hours of battery powered operation between recharges. More information on this digital device can be obtained from the company.



Detente: Russian cartoon, Yankee caption

"Vyalesheff, you have only one chance. He might blow a fuse!"

COMPUTER GAMES
OF OTHER SORTS

(Including Computer Checkers, Computer Go, Computer GOMOKU, Computer Word Scramble, Computer Poker, Prisoner's Dilemma, etc. Submissions of these and other "intelligent" games welcomed by this department. Address all correspondence to COMPUTER GAME DEPARTMENT, (Personal Computing.)

The INTERIM 2 program by Walter Reitman and Bruce Wilcox of the University of Michigan is the result of an effort to construct an intelligent GO playing program modeled upon pattern recognition and problem-solving capabilities of highly skilled human GO players. The enormous task in creating such a program is seen from the following comparison table:

- 10^{11} = Number of stars in the galaxy. (100 billion)
- 10^{40} = "Number of possible choices of moves in checkers which, at 3 choices per millisecond would still take 10^{21} centuries to consider" — Dr. A.L. Samuel.
- 10^{90} = "Number of years required to calculate the first move of chess on a computer operating at a rate of one variation per microsecond." — Claude Shannon
- 10^{110} = Number of atoms in the universe (Einstein).
- 10^{120} = Number of possible positions in chess.
- 10^{761} = Number of possible positions in GO.

What follows is the complete record of a game between the INTERIM 2 version of the Reitman and Wilcox GO program and F.B., a human player ranking 15 kyu at the time of the game (3/13/78). The program played black and was given a nine-stone handicap. Black plays first. Black's first move was to lay down the nine-stone handicap at D4, D10, D16, K4, K10, K6, Q4, Q10 and Q16. The program won by nine points, having 54 points of territory and 11 prisoners to white's 55 points of territory and 1 prisoner.

	White	Black
1	017	R14
3	F17	C14
5	O15	G3
7	C6	C4
9	E6	F4
11	N4	R6
13	R12	Q12
15	R13	Q13
17	R10	R9
19	Q14	R11
21	S11	Q11
23	R15	S14
25	S13	S15
27	R16	Q15
29	P14	R17
31	Q17	Q18
33	T14	S16
Black at 34 takes off White's R15 and R16		
35	D17	C17
37	C12	D12
39	C11	D13
41	D11	E11
43	C10	E10
45	E16	C16
47	C13	F12
49	D14	E15
51	E13	E12
53	C15	B14
55	B15	D15
57	B13	A14
59	A13	E14
Black at 60 takes off White's D14		
61	F15	Q3
63	S10	S12
65	T12	S9
White at 65 takes off Black's S12		
67	T10	T9
69	Q9	D8
71	C8	E7
73	D7	E8
75	F6	O18
77	N18	P18
79	P17	N17
81	N16	M17
83	L16	L17
85	K17	M16
87	L18	M18
89	L15	M15
91	N15	P16
93	O16	L14
95	K15	M12
97	M14	N19
Black at 98 takes off White's N18		
99	L11	K14
101	J16	M13
White at 101 takes off Black's K16		
103	N14	K13
105	N13	H12
107	N12	L12
Black at 108 takes off White's L13		
109	P9	R7
111	F14	F13
Black at 112 takes off White's E13		
113	F7	J15
115	K16	H16
117	H15	J14
119	H17	N10
121	P10	O11
123	P11	N2
125	N6	G9
127	B4	B3
129	B5	H7
131	H6	K7
133	K6	J7
135	L7	J6
137	K5	H5
139	J4	J5
141	K3	L4
143	L3	L5
145	L6	J3
147	J2	H3
149	M4	H4
Black at 150 takes off White's J4		
151	M5	J4
153	M2	O3
155	N3	O2
157	H2	G2
159	M9	M10
161	D18	C18
163	B2	C2
165	A3	C3
167	K9	J9
169	K8	L19
171	K18	K19
173	J19	M19
175	H18	M1
177	L1	N1
179	R8	T7
181	S8	S6
183	S7	D19
185	E19	C19
187	F18	H1
189	J1	G1
191	K2	B1
193	A2	G6
Black at 194 takes off White's H6		
195	R5	Q5
197	R4	R3
199	Q6	P6
201	Q7	S4
203	P7	O5
205	O6	P5
207	J8	G7
209	F8	L9
211	L8	L10
213	N9	T15
215	T13	P3
217	H14	C9
219	B9	D9
221	B7	O4
223	G8	H8
225	J10	H9
227	D5	N1
229	O12	E3
231	G13	E4
233	G12	G11
235	H13	H11
237	J13	L13
239	J12	J11
241	K12	K11
243	F9	G10
245	F10	O10
247	O9	G5
249	T8	T6
White at 249 takes off Blacks R9, S9 and T9		
251	E5	F11
253	C5	E9
255	A4	F5
257	A1	C1
259	P15	N5
261	S18	R18
263	A15	D14
265	PASS	PASS

The specimen game was played March 13, 1978, between the computer program INTERIM 2, and a human player identified only as "F.B." The plays of each side are numbered in proper playing order. Bill Mann, a 2-dan player of Boston, Mass., has studied the documented game and has added his analysis. "As is common in games between players of different strengths," he comments, "Black (computer) took a handicap of nine stones. This handicap is the largest commonly given, and is said to be worth about 140 points to Black.

"To get an idea of how strong the program is, one should realize that White is ranked at 15 Kyu, which places him near the bottom of the ranks of serious GO players, perhaps equivalent to class D in chess. If the nine-stone handicap is justified, and it seems to be, judging from this one game, the computer would rank 24th Kyu. On the other hand, the computer's general level of play is higher than 24th Kyu. Its strong points are solid play and good defense. Its weak points are very bad end-game play, insensitivity to long range attacks, and a tendency to occasional blunders. These blunders may be due to bugs either in the program itself or in its database of patterns.

"Peeking into my crystal ball, I would not be surprised if the program improved very rapidly and, overall, I'm very impressed. Even the kind of mistakes this program makes are basically the same kind that a typical human player makes when first learning the game. Up until the end-game I might have believed that this was a normal game between two humans, in which white had given black much too large a handicap.

"My comments are directed mostly at black, but that doesn't mean white's play was really much better. Anyway, white usually has trouble finding good moves against such a large handicap, and his excesses must generally be excused."

Analysis of Game

by Bill Mann

Figure 1. (1-46)

White initiates play with a normal-looking series of probes against black's handicap stones in the corners. Black's responses are conservative and solid; in fact, too conservative for best play.

White 13 starts an ill-fated invasion, which black (partly due to white's mistakes) utterly crushes, gaining about 25 points.

Black 20:	a blunder, should be at the point where white played W23
Black 22:	capturing one stone (W17) is better
White 25:	must be at B28, which would take the corner for white
Black 26:	?! B28 would be more straight-forward (even though white may be able to live in the corner)
White 29:	another blunder, W30 gives black bad problems
White 31, 33:	these moves help black; the white group is still dead, and black has the corner as well

White 37 starts the same kind of invasion on the opposite wing, but this time with better results, due to black's cautious play (B40 & B44)

Black 46:	should be at D-18

Figure 2. (47-69)

White 47:	an over-play, which should be answered at C-9 or B-14
Black 50:	misses the mark — should be at E-14, or better B-14
White 57:	play elsewhere. Much later a play at B58 can be used to save the two white stones. This was another defeat for white, who must now protect against a black cut at C-9. But instead white tries to save his dead group.
Black 64:	utterly bad — S-9 kills the white group and is the obvious move. Is this a program bug? Black loses about 25 points.
Black 68:	better at W69 to prevent the cut
White 69:	good (in a 9-stone game). Black ignores this move too long

Figure 3. (70-119)

Black 70 starts an inept attack. Normal procedure is to play first at C-9. The difference is about 20 points.

Black 72:	should be at W73

Black 76 starts a strange attack, which white mishandles.

White 79:	a blunder which costs white about 10 points — better at M-17
Black 80:	a good cut
Black 88, 90:	too small — try J-17
W91, B92, W93:	silliness
Black 94:	bad — gives up the good point at J-17
Black 106:	should be at N-12 to prevent problems later
Black 108:	the last chance for black to protect himself, perhaps at 0-11
White 109:	good — should expect to capture three stones (B18, 66, 68)
Black 110:	a blunder, protecting the wrong group — must be at 0-11

It is now worth about 60 points for either side to play at 0-11. Moves 111-119 are nonsense because they ignore 0-11.

Figure 4. (120-169)

Black 120 was the last chance to play 0-11 and capture white. White 121 captures black instead. Black 124 starts an attack on the single stone at W11. Unfortunately, black mishandles the attack, giving up over 20 points. This is a perfect example of an opportunity to build territory while attacking, but black seems to accomplish this goal only by accident.

Black 124:	too low — try M-3
Black 126:	too small — play the attaching move
Black 130:	an excellent long range attack at K-7, but apparently that's not why black played here
Black 132:	should be J-6
Black 134:	should be L-6
Black 138:	should be at W139
Black 144:	should be at W145
Black 148:	should be at B150
White 149:	should be at W151

Black 150:	should be at W151
Black 154:	should be at W155
White 163:	this shouldn't work — try W165 directly
Black 164:	Black can cut at W165 to punish white for W163

Figure 5. (170-206)

The end-game could be said to start here. Black appears to be about 50 points ahead with normal play. But this is where the computer's game seems to collapse.

Black 170:	this play gives white about 6 points. Play instead at W171
Black 176:	much too soon
Black 180, 182:	needlessly gives white 10 points. A bug?
Black 194:	too small

White 195 starts a ridiculous attack which black defends well up to Black 202; there a play at W203 would save 3 points.

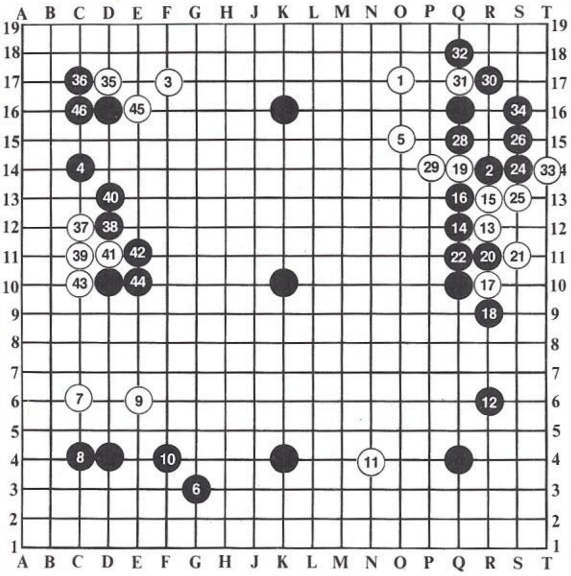
Figure 6. (207-264)

Here at the very end of the game (the easiest part to play well), the computer continues to play quite poorly, giving about a point per move on the average.

Black 208:	pointless
Black 210:	poor
Black 216:	terrible! A bug?
Black 218:	good
Black 222:	poor
Black 228:	provokes W229, giving away 4 points
Black 230:	poor — try F-5
Black 232:	another bug
Black 236:	poor— try J-12 or even J-13
Black 238:	awful
Black 244:	bad
Black 248:	gives up a point

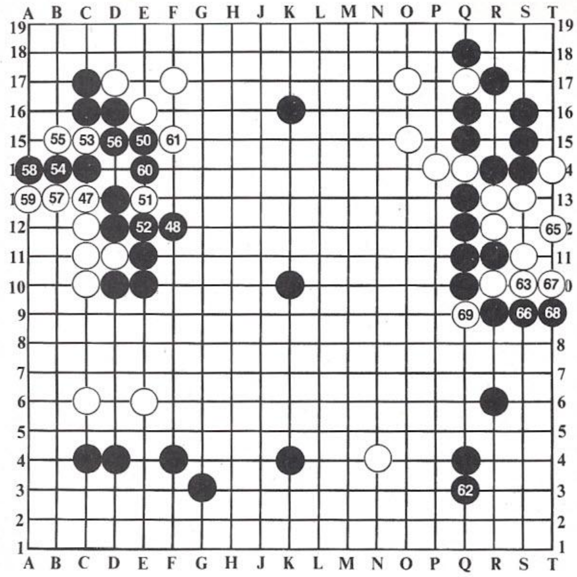
"Fortunately for the computer, the game concluded at this point, leaving it ahead by 8 points. Still, a win is a win, and with a little more work I would not be surprised if this program turned into a competent player. My congratulations to its authors." — Bill Mann

Diagram No. 1



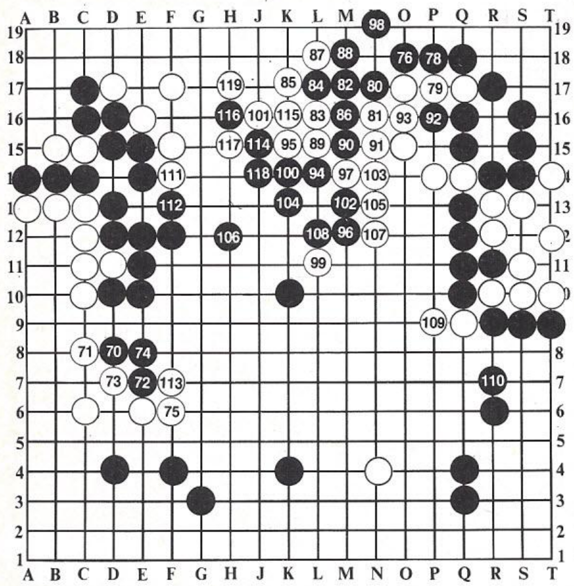
Moves 1 to 46
(Black at Move 34 takes off White's R15 and R16)

Diagram No. 2



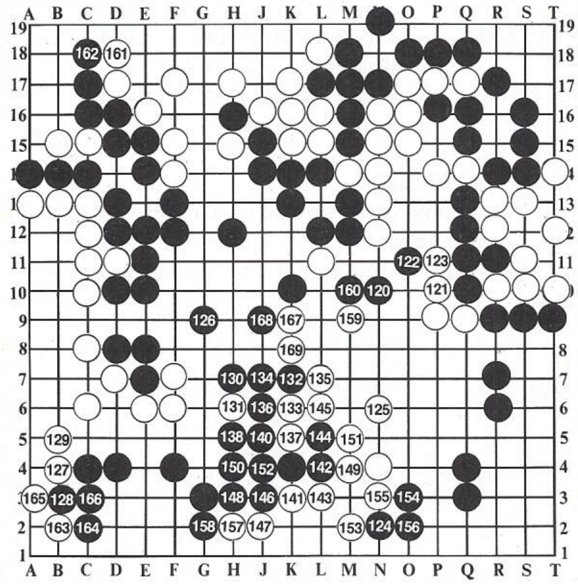
Moves 47 to 69
(Black at Move 60 takes off White's D14)
(White at Move 65 takes off Black's S12)

Diagram No. 3



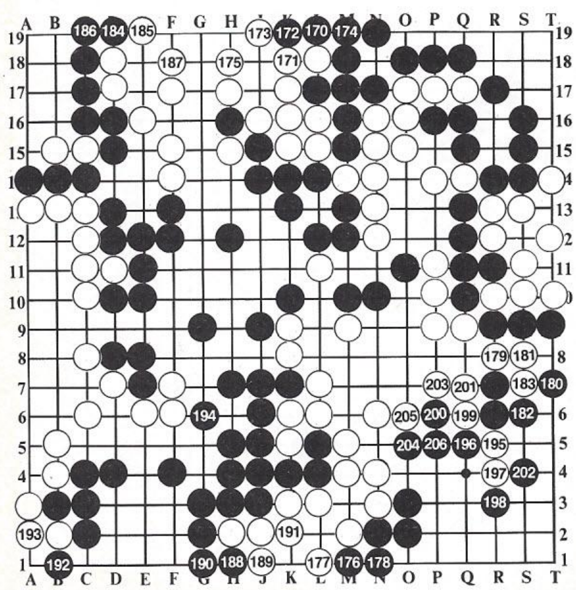
Moves 70 to 119
(Black at Move 98 takes off White's N18)
(White at Move 101 takes off Black K16)
(Black at Move 108 takes off White's L13)
(Black at Move 112 takes off White's E13)

Diagram No. 4



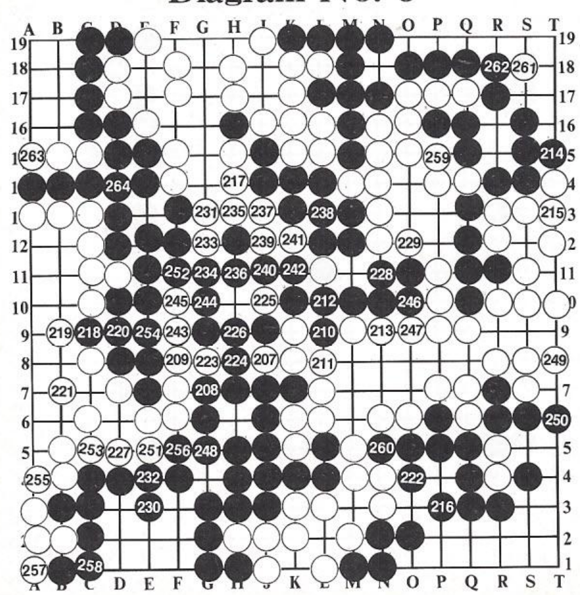
Moves 120 to 169
(Black at Move 150 takes off White's J4)

Diagram No. 5



Moves 170 to 206
(Black at Move 194 takes off White's H6)

Diagram No. 6



Moves 207 to 264
(At Move 265 both players pass to end the game.)

Playing Against the Duisman Program

By Thomas A. Throop

Last month's column considered the computer program of George Duisman for playing bridge with versions available for Commodore's PET, Radio Shack's TRS-80, and the Apple. The program defends while you play as declarer the contract of your choice after you have seen your hand and your partner's. The deals discussed last month were deals 1, 5, 7, 11, and 15 of set 3.65. I suggested for this month that those of you who were interested might play deals 17, 18, 21, 25, and 28 of set 3.65.

Here are my results and you can compare them to your own. On deal 17, I played 4 spades, making 4. On deal 18, 7 spades is obviously the correct contract, which makes easily. On deal 21, I played 4 spades, making 5. I found deal 25 the most interesting, and I shall discuss that one in some detail. On deal 28, I played 4 hearts, making 5.

Returning to deal 25, your cards (South) and your partner's (North) are as follows:

NORTH
(Dummy)
♠ Q4
♥ KJ8753
♦ 32
♣ J86

SOUTH
(You)
♠ A1062
♥ A9
♦ AJ
♣ K7532

I decided to play 4 hearts. Although it looks like an ambitious undertaking, it is, nevertheless, a plausible contract. If you have the Duisman program, why don't you choose a contract (or take mine and play the deal,) before reading further?

Against 4 hearts, the computer West leads the 9 of spades. This hand poses many problems, including locations of the king and jack of spades. I played the 4 of spades from dummy. If East now played the jack, I planned to win with the ace, then lead a spade to the queen, losing to the king. However, this sets up the 10 of spades for the discard of a losing diamond or club from dummy. East, however, played the king on the 4 at trick 1, apparently not holding the jack.

This was a good start. Now, I hoped to find the queen of hearts and planned on losing only two clubs and a diamond. I played the ace of hearts, to be followed by the 9 and a finesse for the queen with dummy's jack. But — on the lead of the ace, East showed out, revealing that West held all of the outstanding hearts. It now looked as though I had to go down two or perhaps three.

I decided not to play another round of hearts just yet. By retaining the 9 of hearts in my hand the defender would have a chance to misplay later. Rather, I planned to enter dummy with the spade queen and then lead a club to my king, hoping East held the ace. Then came more bad news — West ruffed with the 4 of hearts! Now, to my surprise, something good happened — West played the ace of clubs! This made (I thought) my king of clubs good.

At trick 5 West switched to the 4 of diamonds. Presuming that East held either the king or queen of diamonds, (or both), I played the ace of diamonds from my hand before realizing that East had played neither the king nor queen, but rather the 10 of diamonds. To recover this lost trick (which actually should have been lost with proper defensive play), I now led the jack of diamonds, giving the defenders a chance to blunder. West won with the queen and sure enough, as hoped, continued with the 6 of diamonds (the king would be less likely to be wrong). This gave me a sluff and ruff. I discarded the 8 of clubs from dummy and ruffed East's 10 with my carefully preserved 9 of hearts.

Now, I saw an opportunity for end-playing West for down only 1. I planned to ruff a spade in dummy with the 5 of hearts, enter my hand with the king of clubs, ruff a black card with dummy's 7 of hearts, and then lead the 8 of hearts to West's 10. West would now be end-played, holding the Q6 of hearts. while dummy would have the KJ of hearts, for down only 1! Well, the spade ruff with the 5 of hearts worked fine. But when I led the jack of clubs from dummy, East played the queen, I played the king, and West ruffed! He then exited with a diamond and I was down 2.

The tableau below shows the play just described, trick by trick:

	Computer West	North (Dummy)	Computer East	South (You)
Trick 1	9S	4S	KS?	**AS**
2	2H	3H	3S	**AH**
3	**4H**	QS	5S	2S
4	**AC**	6C	4C	2C
5	4D?	2D	10D	**AD?**
6	**QD**	3D	5D	JD
7	6D?	8C	9D	**9H**
8	7D	**5H**	7S	6S
9	**6H**	JC	QC	KC
10	8D	**7H**	8S	3C
11	10H	**KH**	9C	5C
12	**QH**	8H	10S	7C
13	KD	**JH**	JS	10S

Although the computer won this deal, the tableau indicates that East's play of the king of spades at trick 1 was incorrect. East held both the jack and king and should have played the jack. I have commented on my play of the ace of diamonds at trick 5. On this trick West should have led the king of diamonds, holding both the king and queen. At trick 7 west's diamond lead permitted a sluff and ruff, but if a diamond was going to be led, it should have been the king or queen.

The entire deal is shown below:

NORTH
(Dummy)
♠ Q4
♥ KJ8753
♦ 32
♣ J86

COMPUTER WEST
♠ 9
♥ Q10642
♦ KQ8764
♣ A

COMPUTER EAST
♠ KJ8753
♥ —
♦ 1095
♣ Q1094

SOUTH
(You)
♠ A1062
♥ A9
♦ AJ
♣ K7532

Next month I will discuss deals 32, 37, 39, 40, and 41 of the same set, 3.65, of the Duisman program. While waiting for the next issue, you might see how expertly you can play those deals, as N-S.

I have appreciated hearing from those of you who have been sending in comments and questions on computer bridge. I should like to pass along some of these to other readers.

Eugene Minahan of 1980 North Atlantic Avenue, Cocoa Beach, Florida 32931, writes "I am glad someone is taking the lead on developing a library of bridge programs. Has anyone gone all the way yet with a bidding and playing program for a single person against the computer?" The "Bridge Challenger", announced by Fidelity Electronics at the January Consumer Electronics Show, will play one, two, three, or all four hands, while human players play the remaining hands. For those hands which it is playing, the computer will bid and then play as declarer or defender. I shall have more to say about the "Bridge Challenger" in the July or August issue of Personal Computing.

Marshall Brewer, M.D. of 5730 Canyon View Drive, Paradise, California95969, writes "Enjoyed your article on computer bridge. This note is to say thanks and to express my interest in this area. I am a physician in private practice. I have played duplicate at our local club for the past 4 years, about once a week, and play a couple of tournaments a year. My interst in computers started about one year ago and I hope to combine these two interests." He continues, "I would like to work on a bidding program using Bridge World Standard. Will enjoy reading more of your articles and exchanging ideas and programs with anyone interested." My second article appearing in the April issue discussed some early efforts known to me in the development of bridge bidding programs. Next month I shall comment further on the bidding program developed by Tony Wasserman and, as mentioned above, the bidding program in Fidelity's "Bridge Challenger".

I have had some correspondence asking about my past affiliations with Winston Riley III in developing bridge programs. For several years, in the past, we jointly researched new game products for the home consumer market. However, we are now proceeding with our own separate ventures. There was some incorrect information in *Personal Computing's* first reference to computer bridge. The statement said that Mr. Riley "was a frequent bridge-playing partner of mine and that he contributed to moulding the algorithm of the bridge playing program." Neither statement is correct. As a matter of information, I have never sat with Mr. Riley at any table at any bridge tournament. Therefore, in reply to several questions, the intelligence logic of the bridge-playing program is entirely my own.

The interest in computer bridge is very gratifying. Please continue to send in your letters of comment, inquiry, descriptions of your own programs and agreement or disagreement with my views. Address all correspondence to Computer-Bridge Department, *Personal Computing*, 1050 Commonwealth Ave., Boston, MA 02215.

Making Your Lifestyle a Money Machine

The Incredible Secret Money Machine, by Don Lancaster; Howard W. Sams & Co., 4300 West 62nd St., Indianapolis, IN 46268, 159 pages, $5.95 paperback.

Don Lancaster's book, *The Incredible Secret Money Machine,* is remarkable not so much for its subject matter — dozens of books tell you how to set up your own business — as for the fact that it's just plain fun to read. With lucid writing and offbeat humor, Lancaster shows you how to make a living doing whatever it is you *really* want to do.

Say, for example, you're heavily into computers. Why slave at your nine-to-five, just to steal a few precious evening hours at your terminal? Why not make your computer the core of your lifestyle, and make your computer expertise pay the bills? You could, for example, set up a microcomputer consulting business, create software, write articles and books, design and manufacture peripherals or kits — or better yet, some combination of these.

Lancaster's central tenet, laid out in boldface type on page one, dictates that, for the Secret Money Machine to work for you, you must be "into a technical or craft trip on a total lifestyle basis." All else follows from this premise.

In a step-by-step, cookbook manner, Lancaster shows you how to turn your lifestyle into a successful business. The first chapter discusses the broad strategies needed to run a successful money machine. Chapter Two gets down to the nitty-gritty tactics. It's here that Lancaster's whimsical sense of humor, straightforward writing style and hard business sense show most strongly. A few examples:

- "A psychic energy sink is any continuing source of bad vibes that makes you mad enough or involved enough to funnel energy and time away from your money machine. Obvious examples are divorces, anything and everything political, bars, lawyers, and, of course, television."
- "Ease, do not jump, into deep water. Always explore some new area in the simplest, cheapest, and most minimum way. Watch out for sharks and U-boats."
- "Have sane business hours. The only insane business hours are 8-5 Monday through Friday. If you are a night person, work at night. If you like watching the chickens get up, have at it. Burst mode or slow and steady. Whatever works, use it."

Subsequent chapters discuss getting started, keeping informed, tax dodges, investments and other matters.

Among the best material in the book is in the two chapters on communications, subtitled "Words" and "Images". These chapters offer a thirty-minute cram course in printed communication — whether magazine article, spec sheet, pamphlet or advertisement. "Words" gives you 16 pages chock-full of the essentials of good writing, while "Images" offers an equally boiled-down intro to line art, camera work, layout and typography.

The Incredible Shrinking Money Machine, filled with nuggets of information, humor and wisdom, is far too rich for a single reading. You'll dip into it again and again, each time fishing up something new.

— *Reviewed by Don Wood*

The BASIC Handbook, by David A. Lien; Compusoft Publishing; P.O. Box 19669, San Diego, CA 92119; 360 pp.; $14.95 + $1.35 postage and handling paperback.

The BASIC Cookbook, by Ken Tracton; Tab Books; Blue Ridge Summit, PA 17214; 140 pp.; $4.95 paperback.

BASIC, while the simplest computer language, is still a confusing mess. From the time it was first developed at Dartmouth College in the early '60s, BASIC has continually spawned new dialects and versions. Each manufacturer offers his own BASIC; and even within a given manufacturer's product line, the BASICs grow and change, dropping some commands and statements and adding others.

Perhaps nothing brings this BASIC Babel closer to home than trying to run that nifty program you saw in your favorite magazine. You'd really like to use the program — but what the heck does STUFF 3000,65 mean? When your computer hits that line, it just spits out SYNTAX ERROR and sullenly waits for your next move. What do you do?

One solution is to look up the offending statement in one of two recent reference works — *The BASIC Handbook* by David A. Lien (author of Radio Shack's Level I user's manual), and *The BASIC Cookbook* by Ken Tracton.

Both books contain alphabetical listings of BASIC words. In addition, *Cookbook* contains definitions of important words that are not themselves part of the BASIC language — for example, *multiple branching, dummy arguments* and *random data files.*

Handbook, according to the publishers, covers over 50 dialects of BASIC, including various versions from such manufacturers as Radio Shack, Commodore, Cromemco, Apple, Exidy, Microsoft, Imsai, Heath, Ohio Scientific, Processor Technology, SWTP and Wang.

Entries include the word or abbreviation; the category (statement, command, function or operator); the definition, telling what the word does and, often, which computers it works on; a test program to see if your computer accepts the word; hints on programming techniques using the word; notes on variations in usage; and cross-references to related words.

But best of all, Lien tells you how to program around the word (if your computer doesn't have it) and still achieve the same result. In our earlier example, a quick glance at the book reveals that STUFF is Digital Group's word insert values into specified memory locations. We also learn that we can achieve the same result with the more common word POKE (or possibly FILL, on a few computers).

Another example: The FIX function, we're told, "removes all numbers to the right of the decimal point. Its operation is similar to the INT function except FIX does not round negative numbers down." Lien goes on to note that you can simulate the FIX function

using ABS, INT and SGN (absolute value, integer and sign) functions:

A = SGN (N) * INT (ABS (N))

Tracton's *Cookbook* includes word definitions, examples of use, sample programs and flowcharts. He also notes variations in BASIC usage, but not as often or as explicitly as Lien. For example, in defining the SAVE command, Tracton says, "After writing a program in BASIC the user can keep the program either on tape or disk by using the SAVE command." He goes on to give additional information, but fails to note that not all BASICs use SAVE. Lien, on the other hand, also defines SAVE, and then tells us to also read the entry for CSAVE.

Neither book is exhaustive, but that's to be expected. The dozens of constantly changing BASIC dialects make an exhaustive dictionary impossible. Also, each book contains words omitted from the other.

If I had to choose only one of the two, I'd pick Lien's *Handbook* — it's oriented toward the more popular microcomputers, and contains specific information on translating from one BASIC to another. But I'd prefer to have both books, side by side, on the shelf next to my computer.

— Reviewed by Don Wood

Chess and Computers, by David Levy, 146 pp., paperback $8.95;
1975 U.S. Computer Chess Championship, by David Levy, 86 pp., paperback $5.95;
1976 U.S. Computer Chess Championship, by David Levy, 90 pp., $5.95; Computer Science Press, Inc., 9125 Fall River Lane, Potomac, MD 20854.

Chess and Computers was first published in 1975. At the outset the author traces the history of chess machines from the fakes of von Kempelen and his imitators to the grand original, built as long ago as 1890 by Torres y Quevado, a Spanish engineer. Torres' machine had the limited function of playing an ending of king and rook against king, and was mechanically powered, being really a robot. It is still extant in the museum at the Polytechnic in Madrid.

The books of the 1975 and 1976 computer chess champsionships are interesting in showing the development of chess programs. Obviously much progress remains to be made, but what has been accomplished is impressive.

It is *Chess and Computers* which seizes our interest. The difference between a computer program and a human chess player is that a program adds and subtracts; a human analyzes. Chess programs perform their tasks by assigning numerical values to given situations (for example, a plus to moves which increase mobility, a minus to those which limit it). A complex set of rules for the assignment of such values has been evolved, each programmer differing as to what rules (heuristics) should be used and what values assigned. Because this depends on the judgment of the individual programmer, it supports the aphorism that no programmer can devise a program that plays better chess than he or she can play. This will hold true up to the point when programs will be written with self learning and self improvement capacity.

A program that plays checkers has been given this self-improvement capacity, and such a claim has also been made for MacHACK, built by Pinkleaf at MIT. However, the claim made for it in an article in Scientific American (April 1975) that with a high degree of probability White wins with 1-P-KR4 is absurd. One need not be a master or even a strong player to realize that either Scientific American has been hoaxed or MacHACK is improperly programmed. The latter seems most probable since the claim is based on a series of games MacHACK played against itself, hardly a sufficient frame of reference from which to draw so sweeping a conclusion.

Levy shows the gradual improvement of programs since Shannon. This development must have been spurred on by Levy's famous 1968 bet that no program could be developed in ten years strong enough to defeat him, a bet which he won in December, 1978. At the time of the bet Levy was an International Master; he is now a Grandmaster. No program has kept up a comparative pace.

Programmers have consistently underestimated or avoided the difficult part of developing strong programs. They have concentrated their efforts on

---

The World Computer Chess Championship (1974), by Hayes & Levy; Edinburgh University Press, 22 George Square, Edinburgh, Scotland; 105 pages; 3.75 pounds net. (Note: This book is also available for $9 from Edinburgh University Press, c/o Biblio Distribution Center, 81 Adams Drive, Totowa, NJ. 07512.)

The title is misleading because this is much more than a mere recitation of games: it contains a history of computer chess, a chapter on concepts of computer chess, a useful glossary of terms, an analysis of the programs taking part in the competition, and a short chapter on the psychology of chess thinking.

The chapter on computer chess history reports the results of questionnaires dealing with people's awareness of chess computer programs. Typical questions asked of travelers at London's Airport were: Will computers ever play as well as masters and if so, by what year; or, is computer chess worth government funding?

The games are run of the mill, with one notable exception: game number 9, CHAOS against CHESS 4.0, which demonstrated that CHAOS was capable of a purely positional sacrifice of material. (This can be fairly said because CHAOS' "lookahead" was five ply and the text continuation was longer). The position in which the sacrifice was made was comparable to what any Class B player would at least consider. Nevertheless, it does represent an historic first in computer play.

The book's authoritative discussion of computer chess play is clear and interesting. It is combined with a glossary which shows how programs operate.

However, the most interesting chapter of all is the one on chess thinking. Here the authors acknowledge that the next step in the improvement of chess programs will come in programming positional analysis. This will be done, suggests the text, possibly by recognition of types of positions, rather than the current expenditure of energy wasted in reviewing, by rote, huge numbers of positions. The book hints that such programming will be achieved by utilizing the services of a team composed of a chess master, a computer programming expert and an artificial intelligence scientist (or computer pedagogue).

—Reviewed by Morris Miller

''look-ahead'' or tactics by building as long a tree as possible and seeking more rapid computers so they could increase tree searching capacity. This is all very well, but chess is a game of strategy, not tactics, and evaluation is the most important consideration, although it is far more difficult to assess and program. Dr. Emanuel Lasker, a former world champion, wrote that an evaluation is an evaluation of an evaluation, and when finally reduced to bedrock it becomes a set of basic assumptions. It is precisely the determination and weighting of these basics that is difficult. Botvinnik, another former world champion, has observed that only strong players can successfully write a grandmaster chess program. He has helped program the Russian world computer chess champion, KAISSA.

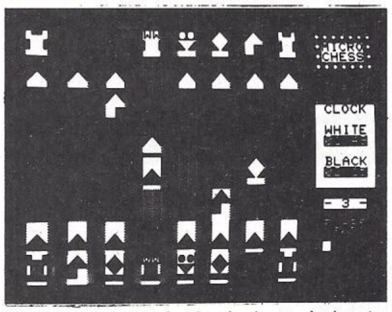
The limitations of the programmer lies between two parameters: It is not feasible to write a program which plays perfect chess by examining every possible variation because such a tree would have to be infinitely long; it is likewise impossible, at least for now, to design an evaluation function sufficiently sophisticated to enable a program to play good chess by examining only a small tree. Because machines are ''stupid'', the solution lies in improving the evaluation function — the most difficult routine to program!

Psychological research has demonstrated that strong players, during the course of a game, consider less than fifty positions as being worthy of further concentrated analysis. This consideration is the equivalent of a program's tree search — but the program undertakes hundreds of thousands of such examinations. Obviously, something is wrong and research, energy and time are being misapplied.

It would seem that Levy's original bet could be renewed for years to come. Perhaps this is true, because how many specialists, named above, could be found to team up for such an arduous task? Nevertheless,this seems to be a shortsighted view because there could well be a ''snowball'' effect of knowledge. This rolling ''snowball'' slowly enlarges, becomes widespread and finally erupts in an avalanche effect. The author shows that today there exists both the potential and the developments in programming out of which could emerge the future world chess champion — a computer program.

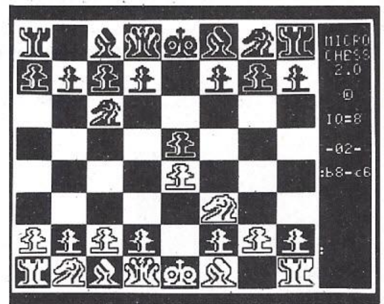
— Reiewed by Morris Miller

PET / TRS-80 / APPLE: Personal Software brings you the finest!

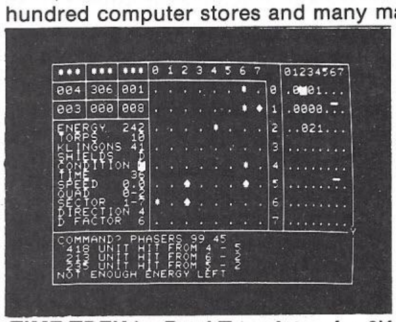


MICRO CHESS

The Industry's First Gold Cassette Over 50,000 Sold

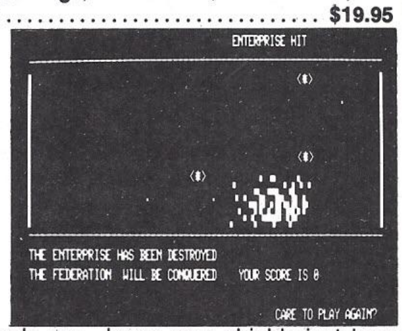


MICROCHESS is the industry's best selling computer game. And no wonder—because MICROCHESS gives you more than just a chessplaying program: A convenient, foolproof set of commands and error checks ... complete instructions in a 5½" by 8½" booklet ... a cassette that's guaranteed to load, with disk versions coming soon ... and several levels of difficulty to challenge you not just once, but time after time. It's available through well over three hundred computer stores and many mail order sources ... always originating from Personal Software. What's more, *every* Personal Software product is selected to give you these same benefits of easy availability, reliable cassettes, readable documentation, a carefully thought out user interface ... and most important, continuing challenge and enjoyment, not just once but time after time. If you haven't already, order your own gold cassette: MICROCHESS, by **Peter Jennings**, for 8K **PET**s, 16K **APPLE**s, and 4K Level I and II **TRS-80s** **$19.95**

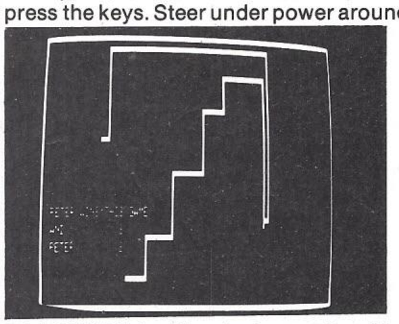


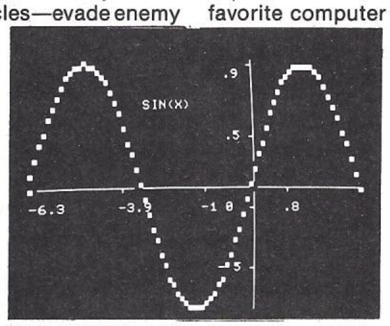
TIME TREK

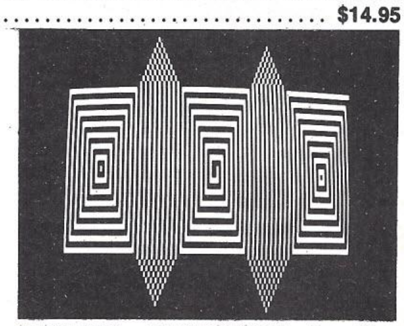
A Tour De Force In Real Time Action Strategy Games



TIME TREK by **Brad Templeton** for 8K **PET**s and **Joshua Lavinsky** for 4K Level I and II **TRS-80s** adds a dramatic new dimension to the classic Star Trek type strategy game: REAL TIME ACTION! You'll need fast reflexes as well as sharp wits to win in this constantly changing game. Be prepared—the Klingons will fire at you as you move, and will move themselves at the same time, even from quadrant to quadrant—but with practice you can change course and speed, aim and fire in one smooth motion, as fast as you can press the keys. Steer under power around obstacles—evade enemy shots as they come towards you—lower your shields just long enough to fire your phasers, betting that you can get them back up in time! With nine levels of difficulty, this challenging game is easy to learn, yet takes most users months of play to master. ADD SOUND EFFECTS with a simple two-wire hookup to any audio amplifier; the TRS-80 also produces sound effects directly through the keyboard case, to accompany spectacular graphics explosions! You won't want to miss this memorable version of a favorite computer game **$14.95**







BLOCKADE by **Ken Anderson** for 4K Level I and II **TRS-80s** is a real time action game for two players, with high speed graphics in machine language. Each player uses four keys to control the direction of a moving wall. Try to force your opponent into a collision without running into a wall yourself! A strategy game at lower speeds, BLOCKADE turns into a tense game of reflexes and coordination at faster rates. Play on a flat or spherical course at any of ten different speeds. You can hear SOUND EFFECTS through a nearby AM radio—expect some razzing if you lose! **14.95**

GRAPHICS PACKAGE by **Dan Fylstra** for 8K **PET**s includes programs for the most common 'practical' graphics applications: PLOTTER graphs both functions and data to a resolution of 80 by 50 points, with automatic scaling and labeling of the axes; BARPLOT produces horizontal and vertical, segmented and labeled bar graphs; LETTER displays messages in large block letters, using any alphanumeric or special character on the PET keyboard; and DOODLER can be used to create arbitrary screen patterns and save them on cassette or in a BASIC program **$14.95**

ELECTRIC PAINTBRUSH by **Ken Anderson** for 4K Level I and II **TRS-80s**: Create dazzling real time graphics displays at speeds far beyond BASIC, by writing 'programs' consisting of simple graphics commands for a machine language interpreter. Commands let you draw lines, turn corners, change white to black, repeat previous steps, or call other programs. The ELECTRIC PAINTBRUSH manual shows you how to create a variety of fascinating artistic patterns including the one pictured. Show your friends some special effects they've never seen on a TV screen! **$14.95**

WHERE TO GET IT: *Look for the **Personal Software**™ display rack at your local computer store. If you can't find the product you want, you can order direct with your VISA/Master Charge card by dialing **1-800-325-6400** toll free (24 hours, 7 days; in Missouri, dial 1-800-3426600). If you have questions, please call **617-783-0694**. Or you can mail your order to one of the addresses below, as of the dates shown.*

Until July 1: P.O. Box 136
Cambridge, Mass. 02138

Personal Software™

After July 1: 592 Weddell Dr.
Sunnyvale, Calif. 94086

CIRCLE 78

WHAT'S COMING UP

SOFTWARE

English-like Language for Small Business Computers

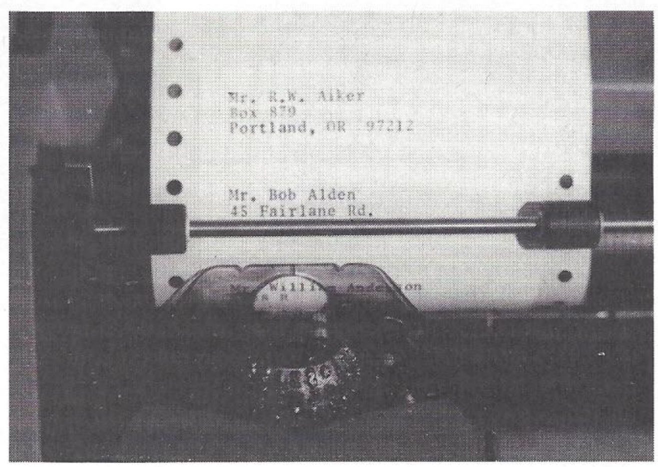
An application development foundation system called Simile provides fully-integrated database management, query and reporting facilities, and is available from Small Business Machines, Inc.

Users require no programming skills to build, operate and maintain systems created with the English-like commands of Simile, the comany said. It provides a range of facilities for data collection and validation, database maintenance, report writing and query. Simile is effective as a stand alone system and as a node on a network, according to the company.

Single installation price of Simile is $6500. For more information contact Small Business Machines, Inc., 527 Madison Ave., New York, NY 10022. *Circle No. 101*

Mailing List Processor Package

The Mailout mailing list processor includes seven modules: Build, Sort, List, Update, Extract, Letter and Help. The package sorts thousands of addresses on Zip or address/title. It merges or extracts subfiles based on codes stored with the addresses. Also it prints envelopes or labels in one or more



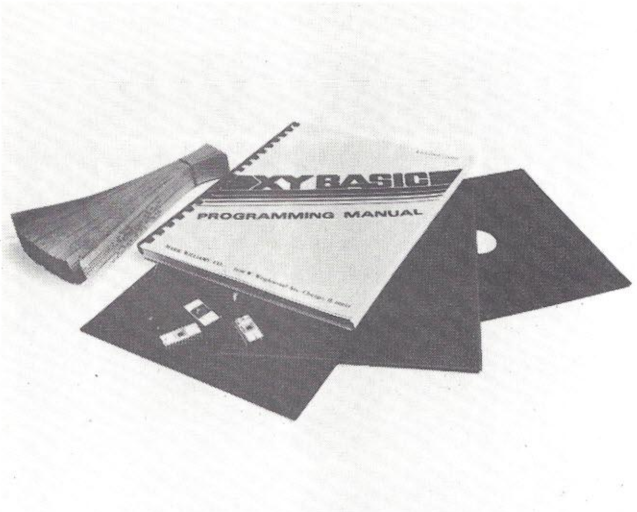
columns and processes letters against mailing lists. Label size is under user control. Mailout is available in three versions: Microsoft BASIC version, Commercial CBASIC and Radio Shack TRS-80 version. It requires dual disk memory.

Mailout costs $6 for the user's manual or $75 for the complete program with disk and source code. State the version desired when ordering. Send a self-addressed, stamped envelope for more information to Center for the Study of the Future, 4110 N.E. Alameda, Portland, OR 97212; (503) 282-5835. *Circle No. 102*

Converting XYBASIC to ROM

The Mark Williams Company introduced the ROM SQuared XYBASIC interpreter, which lets both XYBASIC and your XYBASIC programs reside in ROM.

The ROM SQuared program lets the user "load and go" automatically on start-up, without using floppy disks or cassettes.



ROM SQuared XYBASIC allows users to build readily accessible stand-alone systems. It can keep more than one program in memory at the same time. An XYBASIC command lets the user switch between programs as necessary. And users can transfer programs from ROM to RAM for debugging.

ROM SQuared XYBASIC provides transcendental and string functions. In addition, a Run-Time/Compiler Package can be used in conjunction with XYBASIC to compress the code and reduce execution time.

Versions are available for all 8080-based systems, including ISIS-II, SBC 80/10 and CP/M. Other system features include the added flexibility that a patchable I/O provides.

ROM SQuared XYBASIC is available for about $295. Purchased separately, the XYBASIC Programming Manual costs $20. For more information contact the Mark Williams Company, 1430 W. Wrightwood Ave., Chicago, IL 60614. *Circle No. 103*

Backgammon-Playing Program

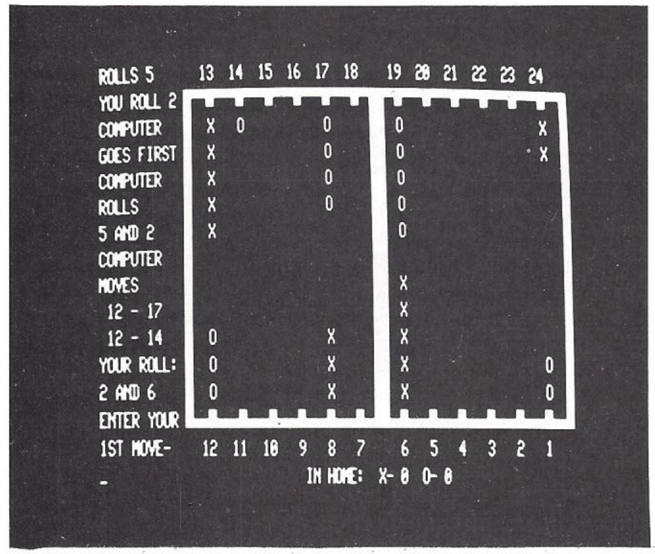
Fastgammon is a backgammon-playing program available for TRS-80 and Apple. According to the game's creator, Quality Software, the computer's decision-making is fast — usually less than a second per move. This is because Fastgammon is programmed entirely in machine language.

Fastgammon displays a backgammon board on the video screen. The computer rolls the dice and displays the roll. Next the player inputs his or her moves and watches as the pieces move from point to point on the board. Then the

computer rolls the dice and responds with its move. On the left hand edge of the screen, the player reviews alphanumeric data and the computer's last move, as well as his own.

One feature of Fastgammon gives the player the option of playing a game using the same dice rolls as the previous game. That is, the same sequence of random numbers is repeated. The player can go back and play a game over using a different strategy to see if the outcome is any different. This can be instructional as well as entertaining.

The computer always plays by the rules of the game, and checks that the player does, too. If a player has no legal



move, the words "can't move" appear in the message area and the computer proceeds with its next move. The input routine makes it difficult to type in the wrong command, and impossible to get out of the program except by hitting the reset button.

According to the company, the computer plays at an average skill level, and because of the luck element involved, it is suitable for players of all levels of skill. Expert players have lost to it, and novices have beaten it, though it rarely loses to rank beginners.

The game, available from retail computer stores, is $20 on cassette or $25 on diskette. An eight-page booklet is included that contains the rules of backgammon, the specific instructions for inputting to the program, and some tips on improving your game. For more information contact Quality Software, 10051 Odessa Ave., Sepulveda, CA 91343.
Circle No. 104

Inventory Control System

Inventory Control, from PolyMorphic Systems, gives System 88 users periodic stock status reports, which are normally expensive to compile manually. Users can create new inventory items and post inventory transactions. Stock status report printouts include total inventory value and value by category.

Each inventory record includes the part number, description, supplier number, data for up to three outstanding purchase orders (date, quantity, P.O. number), weighted aver-

age cost, re-order point, date of last issue, date of last receipt and balance on hand. The system allows 600 inventory records. Add-on storage up to 2400 records is available.

The Inventory Control System is priced at $195 for both manual and diskette. For more information contact Poly-Morphic Systems, 460 Ward Dr., Santa Barbara, CA 93111; (805) 967-0468. *Circle No. 105*

Programs for PET

Softside Software sells four PET Programs: one assembler and three games.

With Assembler 2001, which follows the standard 6502 set of machine language mnemonics, users can write machine code programs, store assembled programs, load them, run them, and list the programs and various PET subroutines. Price is $15.95.

Pinball combines PET's graphic features with the arcade game. Included are bumpers, chutes, flippers, free balls, gates and a jackpot. Price is $9.95.

Super Doodle converts the screen into a sketch pad. Moving the cursor in eight directions leaves a trail of any of PET's 256 characters. Features include an erase key that automatically remembers the last five moves, a return-to-center key and clear control. Super Doodle costs $9.95.

Finally, Driving Ace offers two racing options. One option simulates an endless road with tight turns and twists. The other places the driver on a crowded Grand Prix track to race against the clock. Price is $9.95.

For more information contact Softside Software, 305 Riverside Dr., New York, NY 10025; (212) 866-8058. *Circle No. 106*

TRS-80 Natural Language System and Other Programs

Cybermate offers four programs for Level II TRS-80 — a natural language operating system, astronomy display, animated graphics and a game.

NLOS/1 (Natural Language Operating System) allows the computer to "understand" information given through simple sentences and to answer questions concerning that information. As an educational tool, the system is a study of artificial intelligence through examination of the program's inner workings and its reactions to math reading problems, the company said. It can also be a tool in teaching English grammar, sentence structure and logical deductive reasoning to students, according to the developer. Instructions are included.

Constellation, based on astrophysical concepts, shows the constellations in the night sky in five different views — north, south, east, west and zenith. Users may "travel" to any star in the night sky and the computer will re-orient the views to show what the night sky would look like on a planet orbiting the chosen star. Users may also build their own star charts by entering coordinates (view, X and Y) and the

distance to the star in light-years. Charts can be saved on cassette tape and reloaded later.

Cartoon allows users to create and run animated pictures on the TRS-80.

Battlestar Galactica simulates the television show and allows users free-form strategy in commanding either the Colonial or Cylon fleets.

All four programs, which come on high-quality cassettes, are priced at $5.95 each. For more information contact Cybermate, R.D. #3, P.O. Box 192A, Nazareth, PA 18064. *Circle No. 107*

Business-Aide Software for TRS-80

Occupational Computing Company offers TRS-80 Business-Aide System programs for use in accounting and management.

Accounts Receivable, Billing and Inventory Control for both finished goods and manufacturing include balance forward A/R with monthly details of charges and payments, sales journal, cash receipts processing, on-line customer profile, summary A/R aging, monthly statements, on-line invoicing, revenue analysis, and inventory status. Price for the finished goods package is $750; $1495 for manufacturing.

Accounts Payable includes voucher register, G/L distribution summary, A/P aging report, cash requirements reports, cash discounts, check writer and check register. Price is $350.

Payroll can be used for multiple pay frequencies, automatic deductions, commissions, bonuses, tips and various financial reports. Price is $350.

With the Client Accounting program, simple client set-ups can be established, along with standard or custom charts of accounts, flexible journal entries, selective journal reports, flexible financial statement generation and cumulative payroll reporting. Price is $1495.

Doctors can use the medical billing software for patient log-in, insurance form generation, aged accounts receivable, daily revenue report and payment receipts processing. Price is $1495.

A radiologist-aide program for $1995 allows many of the processes used in the medical billing program.

Other programs include trucking-aide for $1995 and construction-aide for $1995.

For more information contact Occupational Computing Company, Inc., 22311 Ventura Blvd., Suite 124, Woodland Hills, CA 91364; (213) 999 1919. *Circle No. 108*

Small Business System with Custom Programming

Better Programming Systems offers a small business system including a microcomputer and customized programming. The $12,000 system includes an Ohio Scientific Challenger II or III microcomputer. As another included feature, a BPS representative will provide on-site training, and customize programs for users' particular applications. Some parts of the country may require a premium for travel expenses, according to the company.

Other features are a formatted screen to display entire sets

of information, and quick report writing. The system is suitable for companies which do $0.5 million to $3 million in business annually, according to BPS. However, large companies with other computers could use the BPS system as support.

Standard applications are Report Writer, Data Update and Data Query. Optional programs include Payroll, General Ledger and Word Processing.

The computer includes one Megabyte mass storage, CRT with full-sized keyboard and a 125 lpm, upper/lower case high-quality printer. Typewriter-quality printer, hard disk, CP/M operating system and other specifications are optional. The system can be upgraded to 300 Megabytes with several data entry and retrieval stations.

For more information contact Better Programming Systems, Inc., 275 Fort Washington Ave., New York, NY 10032; (212) 765-0815. *Circle No. 109*

Accounts Receivable and Payroll for TRS-80

National Software Marketing Inc. announced two Inflation Beater products for the Radio Shack TRS-80 computer. A payroll package and an accounts receivable package are written for a DOS level II system with two disks and a printer.

The Payroll handles 99 employees with monthly, semimonthly, weekly and bi-weekly pay periods. There are two hourly rates per employee with overtime equaling 1½ times the first hourly rate. The program contains a master file maintenance module that permits entering new employees, changing employee data, zeroing QTD data and zeroing YTD data. It also permits adjustments for normal checks. Payroll produces a current period check register, quarterly report and employee status reports.

Accounts Receivable, including shipping and handling, is designed to replace McBee accounts receivable procedures. An in-line credit check capability allows for 8000 accounts receivable journal, credit check posting report, MTD transactions and totals, master file listing, printed statements and trial balance. The system uses a balance forward accounting, computes 1½% service charges for each account, and produces 4-line label data for name/address in master file for each account.

Each package is $55. For more information contact Elliot B. Kleiman, National Software Marketing Inc., 4701 McKinley St., Hollywood, FL 33021; (305) 625-6062. *Circle No. 110*

Ugly BASIC Answers

The hidden word in Mr. Cook's "Human BASIC" program is ATTENTION. The Ugly BASIC program prints out the word COMMITTEE. If you have a BASIC puzzle you'd like to share with our readers, send it to Feedback Editor, *Personal Computing*, 1050 Commonwealth Ave., Boston, MA 02215. — *The Editors*

WHAT'S COMING UP

Sixteen Personal Programs

Aladdin Automation announced 16 Personal Programs in two releases of eight programs each.

Each of the 16 programs are designed for the TRS-80, PET 2001 and Apple II personal computers.

In the first release, Math-Ter-Mind gives pre-school children the opportunity to learn simple mathematics and have fun with it. Included in the program is a face of a Mathematician, which breaks into a smile when given a correct answer. Lunar Lander takes you through the thrills and the dangers of a landing on the moon, the company said. Craps simulates the popular casino game.

Also in the first release, Jungle Island leaves the player stranded on a seemingly deserted jungle island. Many dangers and treasures are to be discovered along the way as the player leaves the relative safety of the beach and enters the jungle. Mastermind tests powers of deduction, reason and logic. The task consists of discovering the computer's secret code, using clues provided by the computer. Stix, an update of the ancient game of Nim, is played with two to five piles of sticks with a varying number of sticks in each pile. The player removes these sticks while competing against the computer. Tic-Tac-Toe, an update of the traditional game, features several levels of difficulty. Super Pro Football recreates evey Super Bowl game ever played, giving players the once-in-a-lifetime chance to change history by going for that one make-or-break play, according to the company.

In the second release of Personal Programs, Stox is a stock market simulation. The Psychologist involves a running conversation with the computer. Multiplication Cave gives elementary school children the opportunity to learn multiplication tables, with each correct answer leading the child one step further through a cave. Home Management gives a running inventory of food and other household items.

Also in the second release, Personal Finance is designed to help with individual finance, such as balancing check books. Tank Battle graphically creates battles between armored tanks. Starship Enterprise features elements of good and evil locked in an outer space struggle. Arabian Nights tells the magical stories of Aladdin and his lamp.

For further information contact Aladdin Automation, Inc., 3420 Kenyon St., Suite 131, San Diego, CA 92110.
Circle No. 111

Mailing/File Program For TRS-80

Tarzac/Computer Products sells the Ultimail Disk Based Mailing/File Label Print Program written by Ron Wagener of Computer Generated Data. Ultimail utilizes Sequential Disk Files and operates under all versions of TRSDOS. Ultimail can pack 1000 files on a system disk in a one drive system depending upon the available RAM. Minimum RAM required is 32K. Features include Hal Program Monitor System which fully protects data against loss by protecting against defective data input; user-controlled spacing between printed labels, which allows the use of various speed-mailer forms and material; user-controlled print quantity, allowing the user to specify how many of each label to print. The three-line labels are printed in ascending Zip Code order.

Self-protecting and user-interacting Ultimail also includes

extensive Edit and Additional Files routines, and uses variables to Name, Save and Load files. Priced at $55 on disk, the program is available from Tarzac/Computer Products, Box 10203, Norfolk, VA 23513. *Circle No. 112*

Converting BASIC Programs to Calculator Programs

Anyone familiar with BASIC can quickly create a complex calculator program without ever needing to know anything about the calculator language by using a new software product from Singular Systems.

This BASIC to TI 58/59 Cross Compiler (translator), written in BASIC, accepts BASIC programs as input and automatically translates them into keystroke programs for Texas Instruments TI 58 and 59 calculators. The user can develop, refine and test his programs conveniently in BASIC, taking advantage of the high level language and BASIC's editing and debugging features. The resulting program can be compiled on the same machine into an equivalent keystroke program ready for immediate use on the user's calculator.

Extended features of the Cross Compiler are keystroke optimization (elimination of redundant operations and unnecessary parentheses); complete keystroke program listing geared to printers or video monitors; listing of BASIC variables used and corresponding calculator memory registers; listing of calculator "labels" used; and recognition of standard and non-standard BASIC commands and functions.

This software is aimed at educational users and hobbyists with access to a BASIC machine with 16K RAM. Price is $65 for BASIC Cross Compiler program listing, including comprehensive user's guide and documentation. For more information contact Singular Systems, 810 Stratford, Sidney, OH 45365. *Circle No. 113*

Word Processor for TRS-80

The Peripheral People introduced word processing programs for TRS-80 single or multiple disk based systems. The Electric Secretary was written in BASIC to permit user customization and requires a minimum memory of 32K.

One feature is a hyphenating dictionary. When long words at the end of a line leave large gaps in the text, the TRS-80 asks the operator to hyphenate the word. The word is then stored in the dictionary with correct hyphenation points and the text is printed, hyphenated and justified.

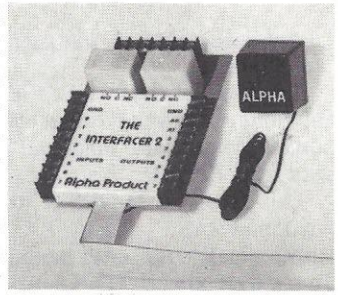
File coupling permits preparing lengthy manuscripts without overloading memory. The program is suited for automatically generating form letters. Address lists and form letters can be cross coupled without operator intervention. The operator can insert text during printout if desired.

In applications for adding "boiler plate" paragraphs to manuscripts and letters, the operator can call up any number of standard files. Features include an echo routine to permit printer use as an electric typewriter and an upper case shift lock (when TRS-80 is modified for upper/lower case).

Electric Secretary, supplied on formatted disk, is priced at $75 postage paid. Upper/lower case conversion information is free on request. For more information contact The Peripheral People, Box 524, Mercer Island, WA 98040. *Circle No. 114*

PERIPHERALS

Interface to Control External Devices



With a new input/output interface by Alpha Product Co., TRS-80 computers can switch or control appliances, and sense switch closures, photosensors or logic levels.

The Intefacer 2 plugs directly into the 40-pin edge connector on the rear of the TRS-80 microcomputer interface. Users control inputs and outputs with simple Level II BASIC INP and OUT statements with eight input channels.

The user's manual includes programs, detailed diagrams and explanations. Sample programs include an automatic telephone dialer and a digital lock.

Two of the outputs are SPDT relays, the other six are TTL level. Up to six more relays can be controlled externally. Two of the inputs are opto-isolated to safely sense varying voltages. All eight inputs will accept either contact closure or TTL level logic. Other families of logic an be easily interfaced, according to the company.

The assembled and tested Intefacer 2 is priced at $85 plus $3 for shipping and handling. Power supply, connector cable and user's manual are included. For more information contact Alpha Product Company, 85-71, 79th St., Woodhaven, NY 11421; (212) 296-5916. *Circle No. 115*

Answers to BASIC without Style

```
EX CHAL1
 3
 4
 5
 6
 7
 8

READY

EX CHAL2
? 3,7
 3  7  2  7  0              1  0  0
? ^P
STOP AT LINE 10
READY

EX FLEE1
******    *         ******    ******    ******    ******
*         *         *         *         *         *
*         *         *         *         *         *
******    *         ******    ******    *         ******
*         *         *         *         *         *
*         *         *         *         *         *
*         ******    ******    ******    ******    ******

READY

EX FLEE2
 111111    1         111111    111111    111111    111111
 1         1         1         1         1         1
 1         1         1         1         1         1
 111111    1         111111    111111    1         111111
 1         1         1         1         1         1
 1         1         1         1         1         1
 1         111111    111111    111111    111111    111111

READY
```

Apple II Light Pen

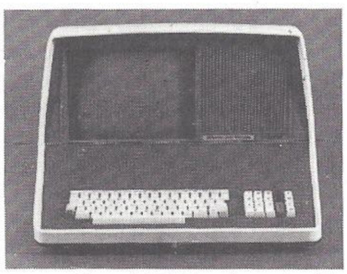
An Apple II light pen is available for applications such as bar graphs, charts and games. The Light Pen is supplied with three demonstration programs on cassette. These demo programs exemplify the uses of the Pen, and aid in developing BASIC programs to drive the pen.

The first demonstration tells how to use the Light Pen as a menu selection tool. Second is a program of graphics demonstrations which permit the user to select from a menu of graphic shapes and colors. Selection from either the shape or color menu is accomplished by depressing the Return key. The third program is a graphics color bit-pad demonstration. A color may be selected from the color menu by depressing any key.

The Pointer software driver performs seven functions, which include selection of graphics mode and page two display, search for x and y ordinates, a test for odd/even y ordinate, set page one display and return to calling program.

The Pen is backed by a 90-day warranty. The entire package, including light pen, software on cassette and operating manual is $34.95. For more information contact Programma International, Inc., 3400 Wilshire Blvd., Los Angeles, CA 90010; (213) 384-0579. *Circle No. 116*

RS-232C Data Entry Terminal



Intertec's new InterTube II video display terminal is suitable for business and school applications, according to the company. The $995 data entry terminal offers upper and lower case characters displayed on an 8 × 10 matrix, a 24 line by 80 character screen, a complete ASCII keyboard with 18-key numeric pad, 14 user-defined function keys and a hooded display to cut down glare and give extra privacy.

A text editing system features character and line insert/delete. The terminal offers full cursor addressing, automatic

repeat for all keys, individual backspace and shift lock keys and a graphics mode for design and display of forms. A text editing system with character and line insert/delete is also included.

InterTube's RS-232C interface operates from 50 to 9600 bps, as does the RS-232C printer port.

For more information contact Intertec Data Systems, 2300 Broad River Road, Columbia, SC 29210. *Circle No. 117*

PET Floppy Disk & Expansion

PEDISK, a combination memory-I/O expander and floppy disk system for the PET, provides both a high-speed floppy disk and an S-100 expansion chassis. The S-100 expansion will hold extra I/O cards and memory boards for addition of a printer, telephone interface, modem voice I/O cards.

The floppy disk is available with up to 3 minifloppy disk drives (total capacity of 80 Kbytes) or up to 4 full size disk drives (total capacity of 1 Megabyte). System prices range from $799.95. For more information contact CGRS Microtech, P.O. Box 368, Southampton, PA 18966.
Circle No. 118

Light Pen for the PET

A self-contained light pen, which adds versatility to graphics programs and games, plugs directly into the Commodore PET 2001 user port.

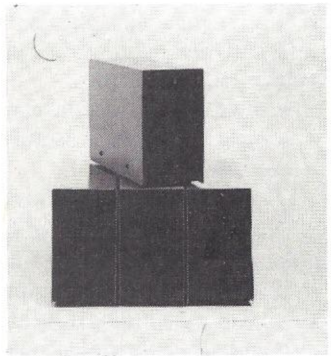
The light pen, from 3G Company allows users to bypass the PET's keyboard and interact directly with information displayed on the CRT screen.

A "MENU" can be displayed on the screen and the user can select from that menu with the light pen. This interaction helps the non-computer oriented person use an applications program.

Another use of the pen is in educational programs. The pen allows a child to interact directly with the screen display and the child need not know how to type.

The light pen is completely assembled and ready to plug into the PET. A sample program and programming instructions are included. The pen is sold mail-order with a thirty-day, unconditional money-back guarantee. The entire package sells for $29.95 (plus $1.50 for postage and handling within the United States; $6 for foreign orders). For more information contact 3G Company, Rt. 3, Box 28a, Gaston, OR 97119; (503) 662-4492. *Circle No. 119*

Add-On Disk Drives for TRS-80



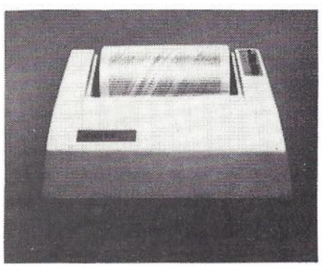
Microcomputer Technology, Inc., manufactures add-on single and dual head disk drives for the TRS-80. The single-head family offers the user a choice of MPI, Pertec or Shugart SA400 mini floppy disk drives. The Shugart is the same device offered by Radio Shack, while the Pertec provides quieter operation and the use of the Flippy diskette (uses both sides), said the company. The MPI unit provides additional features normally found in the larger 8" disk drives, such as door lock and automatic diskette ejection, said MPI.

Drives interface to the TRS-80 via the Radio Shack TRS-80 expansion interface, which can accomodate up to four single-headed drives or two double-headed drives. Operating software is available from Radio Shack.

The dual headed units provide the same capacity as two single-headed drives.

Prices for the single head units start at $379. The dual-headed units are priced at $675. For more information contact Microcomputer Technology, Inc., 2080 S. Grand Ave., Santa Ana, CA 92705; (714) 979-9923. *Circle No. 120*

Low-Cost Printer



A new low-cost, non-impact printer from Computer Printers International (Comprint) is designed to meet the needs of small business system and home computer users.

Alphanumeric images in full 96-character ASCII set, with upper and lower case, are delivered by the Model 912. The print medium, a 9 × 12 printhead matrix, generates overlapping dots to create fully formed characters.

The unit writes 80-column lines at 225 characters/second (170 lpm) on 8½" wide paper. IEEE-488 and strobe/acknowledge are supplied with the parallel-interface model; RS-232C and 20 current loop versions are available on the serial-interface model priced at $39 more.

This relatively high speed amounts to about 3-typewritten pages per minute. At that rate, a 300-page-novel can be turned out in about an hour. Cost of the aluminized paper is $8.00 for a 300 foot roll, equivalent to 300 sheets of 8-1/2 × 11 paper. Is this more expensive than impact printing? Harlen Dybdahl, vice-president of marketing, says it is much cheaper. A 300 foot roll of IBM paper, he says, costs about $2.75. To this must be added the cost of short-lived ribbons, type faces, etc., an expense of about $15, bringing the total cost of impact-printing to about $17.75 for the 300 foot roll.

Meanwhile, Comprint's cost for the same roll remains at $8. Any argument against the use of aluminized paper disappears, says Dybdahl, when it is shown that the paper produces clear photostat copies.

The price of $596 for the parallel-interface model makes it an attractive buy, and a useful addition to any computer system, adds Dybdahl.

Of special interest to hobbyists who work only from the CRT display is the ability of Comprint to produce "CRT hardcopy." Any data appearing on the display of a CRT can be transfered to the Comprint in a moment or so. This allows the printer to be used as backup to main printers. If only a small section of data is needed from a CRT without waiting for the main printer to finish its run, the desired information can be quickly transfered to the Comprint and, after photostatting, can be delivered,mailed or transmitted to any destination.

The printer weighs under 15 pounds and measures 15¼"W × 13½"L × 5¾"H. The Model 912 may be purchased through computer stores.

For more information contact Comprint, 280 Polaris St., Mountain View, CA 94043; (415) 969-6161.
Circle No. 140

COMPLEMENTS

Device to Solve TRS-80 CLOAD Problems

The Peripheral People introduced the Data Dubber to eliminate problems in CLOADing TRS-80 program tapes.

According to the company, TRS-80 tapes are produced on high-speed duplicators. Some distortion is introduced by these machines. Recordings with waveform distortion, noise, hum and minor dropouts can be regenerated by the Data Dubber to produce data pulses identical with the TRS-80 CSAVE data stream. These idealized pulses can either CLOAD the TRS-80 or feed a second recorder for duplicating tapes. Because the recorder only copies pulses, data and machine language tapes can be reproduced as easily as those in BASIC. The regeneration technique works equally well on Level I or II recordings.

With an electronic switch activated by the input of data pulses, users cannot forget to turn off the nine volt battery.

Data Dubber is priced at $39.95, postage paid. The company offers a money back guarantee of satisfaction. For more information contact The Peripheral People, Box 524, Mercer Island, WA 98040. *Circle No. 141*

Apple II & Sorcerer 16K Up-Grade Kits

Ithaca Audio expanded its line of high density 16K memory expansion kits with two more Simple Up-Grade Kits. The original TRS-80 Simple Up-Grade Kit is now joined by the Simple Apple II and the Simple Sorcerer Up-Grade Kits.

Each complete kit includes eight prime tested 16K RAMs plus the proper preprogrammed jumper shunts or memory

units. Anti-static foam in the package protects all components. Installation instructions are included. All parts are pre-tested and guaranteed for life, according to the company.

For more information contact Ithaca Audio, Box 91, Ithaca, NY 14850; (607) 257-0190 *Circle No. 142*

P.C. BOARDS

Single Card Computer

Pragmatic Designs introduced CPU-1, an 8085-based microcomputer designed specifically for dedicated control applications. CPU-1 allows manufacturers or hobbyists to concentrate on their overall application, rather than on detailed microcomputer design,the company said.

CPU-1 contains a 3 MHz 8085 microprocessor. The basic system contains 256 bytes of RAM, 22 I/0 lines, one serial I/0 port and one programmable counter/timer. The system includes an RC clock, as well as an optional crystal for users with precision timing applications. The system has both power on reset and a manual reset button, and the 8085's vectored interrupt structure is supported.

The CPU-1 card has sockets for 1 to 4K of EPROM. The EPROM type is jumper selectable to allow use with2708, 2758, 2716 or TM2716 EPHROMs.

For users with larger applications, the CPU-1A offers all the features of CPU-1 expanded to 512 bytes of RAM, 44 I/0 lines, and 2 programmable counter/timers.

CPU-1 has a special PC board area laid out for wire wrapping or soldering special user functions. System I/0 and control traces are bussed to pads adjoining this area, allowing users to customize the CPU-1 for their specific applications according to the company. This often eliminates the need for any other system boards. CPU-1 also has space for an on-board power supply with plus or minus and plus 12V outputs. A power supply option, including everything but transformers, is available.

Applications programs for CPU-1 can be developed using 8080/8085 development systems. CPU-1 is also designed to work with Pragmatic Designs' DBM-1 debug memory card. A connector and logic to utilize the DBM-1's hardware trap facility is provided and a trap LED shows the trap latch state during debugging. This feature allows programs for CPU-1 to be developed quickly using a larger S-100 computer, the company said. Once development is complete, the EPROMs can be programmed and installed in CPU-1 for normal system operation.

CPU-1 is available for the following prices: CPU-1 (256 bytes RAM, 22 I/0 lines, one timer) $125 kit, $175 assembled and tested; and CPU-1A (512 bytes RAM, 44 I/0 lines, two timers) $160 kit, $210 assembled and tested. Option prices for either CPU-1 or 1A: crystal $10; power supply (less transformers) $20 and interface connectors $10. OEM and quantity discounts are available, with distributor and dealer inquiries invited. For more information contact Pragmatic Designs Inc., 711 Stierlin Rd. Mountain View, CA 94043; (415) 961-3800. *Circle No. 143*

WHAT'S COMING UP

32K RAM Memory Board

Tarbell Electronics announced its new 32K Static RAM board. The new memory board is S-100 bus compatible and runs at 300 ns. It features extended addressing, or bank switching, and contains nine regulators designed to enhance its heat distribution.

The board is fully factory assembled and tested prior to shipping . The same board is available with only 16K.

The Tarbell Memory Board is supplied with a 20-page operating manual which includes a complete parts inventory, schematics and several test routines. Backed by a one-year warranty, the 32K RAM Board is priced at $625; the 16K version costs $390.

Both boards may be ordered by mail with accompanying payment, plus 6% sales tax for California residents, from Tarbell Electronics, 950 Dovlen Place, Suite B., Carson, CA 90746; (213) 538-4251 or 538-2254.

Circle No. 121

ADVERTISERS' INDEX

SALES OFFICES

Northeast: George Palken, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic:** Arthur A. Daks, Arlene Joseph, Benwill Publishing Corp., 92 S. Central Ave., P.O. Box 702, Valley Stream, NY 11582; (516) 872-9550 • **Midwest:** Hank Bean, 2663 Hillside Lane, Evanston, IL 60201; (313) 475-7173 • **Northwest:** Ralph Peterson, 1380 Oak Creek Dr., #212, Palo Alto, CA 94304; (415) 328-4392 • **California:** Benwill Pulbishing Corp., 924 Westwood Blvd., Los Angeles, CA 90024; (213) 478-3017 • **Japan:** Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-ku, Tokyo 166; (03) 311-1746.

CLASSIFIEDS

MICRO MANAGEMENT SYSTEMS

MICRO-COMPUTER SPECIALIST
LARRY OWENS

● 15% DISCOUNT-TRS-80's ●

RADIO SHACK DEALER - MINI MALL
DOWNTOWN SHOPPING CENTER
CAIRO, GA. 31728

912-377-7120

TRS-80 Quality Software

Diskette:	word processor	$39
	mailing list	$35
	data base	$49
	inventory	$39
	key random-access util	$19
Level II.	word processor	$29
	sort & list access util	$16
Level I or II:	inventory	$20
	cassette data base	$20
	check balance & security	$15

Poi Pow
96 Dothan St.
Arlington, MA 02174

TURNKEY DUAL DISK BUSINESS SYSTEMS NOW STARTING AT UNDER $4000 WITH OVER A MEGABYTE OF MASS STORAGE AND SEVERAL DOZEN FIELD-TESTED APPLICATIONS PACKAGES TO CHOOSE FROM

Let New England's largest computer store serve your data processing requirements!

We have over 4000 square feet devoted to the display, sales and service of computer systems and equipment in our Waltham location just off Route 128 exit 49, two miles north of the Mass Turnpike outside Boston. Send 50¢ to cover the mailing cost of our new catalog. We stock the widest selection of new hardware and software of any New England computer store.

Computer Mart, Inc.
1395 Main Street
Waltham, MA 02154
(617) 899-4540

Save More Than 20%!

NORTH STAR — XITAN
The smartest computers at the smartest price
DOUBLE DENSITY

HORIZON-1-16K kit, list $1599	**$1279**
Assembled & tested, list $1899	**$1519**
HORIZON-2-32K kit, list $2249	**$1799**
Assembled & tested, list $2549	**$2039**

Powerful NORTH STAR BASIC - FREE
XITAN Computers — NOW —
With QUAD DENSITY DISKS!
Famous XITAN Software & Basic - **FREE**
INTERTUBE II Terminal, list $995 **$795**
Which Computers are best? BROCHURE - **FREE**
AMERICAN SQUARE COMPUTERS
KIVETT DR. JAMESTOWN NC 27282
(919) 883-1105

FOR **TRS-80™** OWNERS
T-PAL

Programming Amateur's Letter

THE "DO-IT-YOURSELF" SOFTWARE NEWSLETTER

Published Monthly, We'll Teach You All the Latest Wrinkles — How to Get The Most out of Your Computer

Graphics • Games • Personal
Home • Business • Finance
And Much More!

$24/year . . . or write for FREE Details

THE MAIL MART
Box 11102P San Francisco, CA 94101

LOW USAGE

TELETYPE® MODEL 33 ASR Computer Terminal

Reduced to $750.
Limited Quantity

- Copy Stand
- Base
- Tape Punch
- Tape Reader
- X-on and X-off
- Line/Local
- 30 day guarantee
- Completely overhauled

D.P.S.I., 60 Hickory Dr.
Waltham, MA 02154
(617) 890-4445

PET COMPUTER OWNERS

NEW line of software for the 8k PET. Each tape contains TEN intelligent games and tests. Hangman II, Quotes, Pro Tic-Tac-Toe, Not so easy, Psyc. Test, Mystery Quiz & more! Utilizes PET's EXCELLENT GRAPHICS! Unconditionally Guaranteed. Send ck. $5.00 ppd. Same Day Service.

VIP ENTERPRISES
P.O. BOX 17011
SALT LAKE CITY, UTAH 84117